Structural Analysis Algorithms for Nanomaterials

Larsen, Peter Mahler

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Structural Analysis Algorithms for Nanomaterials

Peter Mahler Larsen
Center for Atomic-Scale Materials Design
Department of Physics
Technical University of Denmark

1st September, 2017

Preface

This thesis is submitted in candidacy for the Ph.D. degree at the Technical University of Denmark (DTU). The work has been carried out from September 2014 to August 2017 at the Center for Atomic-Scale Materials Design (CAMD) and the section for Neutrons and X-rays for Materials Physics (NEXMAP) at the Department of Physics, under the supervision of Jakob Schiøtz and Søren Schmidt.

I thank my supervisors for taking me under their wing as a scientific apprentice and teaching me their craft. Learning to investigate an idea, and take it from an interesting observation all the way to publication has been a rewarding experience; thanks to them, I have the confidence that I can now do so independently. I am also grateful to my colleagues in both the sections in which I carried out my studies, the scatterers in NEXMAP and the computers in CAMD, for creating a friendly and open learning environment. In particular, I would like to thank Karsten Wedel Jacobsen for countless fascinating discussions, which have resulted in a couple of papers.

For hosting me during a fantastic external research stay, I would like to thank Christopher Schuh and the members of his group at MIT. The collaboration with fellow Ph.D. student Arvind Kalidindi on cluster expansion models has been especially fruitful. Also to Alexander Stukowsi and Karsten Albe for hosting me at TU Darmstadt. Without Alex having integrated the PTM method in his OVITO software framework, it would not have reached an audience outside DTU. I thank Emanuel Lazar for many discussions and for inviting me to the SIAM Conference on Mathematical Aspects of Materials Science. Meeting and becoming part of the scientific community has been one of the most rewarding experiences of my studies.

Marianne Ærsøe, Hanne Sørensen, and Helle Ingleby Jørgensen deserve much gratitude for their endless patience, as do Ole Holm Nielsen, Bernd Damman and Sebastian Borchert for their help on all matters supercomputing-related. Lastly, I would like to thank Henning Bo Nicolajsen for help with administrative matters and Jane Hvolbæk Nielsen for guidance.

Copenhagen, September 2017
Peter Mahler Larsen

Abstract

This thesis presents a reformulation of existing problems in materials science in terms of well-known methods from applied mathematics: graph theory, computational geometry, and mixed integer programming.

The centrosymmetry parameter is reformulated as a graph matching problem, and resolves the inconsistencies in the existing calculation methods as a consequence.

By formulating the distance function of lattices as a bipartite graph matching problem, it is shown that the similarity between crystal lattices (root mean square distance, RMSD) can be calculated in polynomial time, which improves upon the existing factorial-time bound. This method is subsequently extended to two-dimensional monolayers.

A method is presented for the identification of ordered crystalline phases in molecular dynamics simulations. A robust classification is obtained by the use of template matching, also formulated as a bipartite matching problem on geometric graphs. This method is adapted for two-dimensional materials, in order that e.g. defect structures in polycrystalline graphene can be studied.

Matrix decompositions are used to develop a geometric lattice matching algorithm, which can exhaustively identify all low-strain interfaces. The stable, low-energy interfaces which are found as a result are intended for use in the design and construction of topological superconductors, which have important applications in quantum computing.

Cluster expansion models are used to find ground-state structures in gold-silver nanoparticles, which are used in a variety of catalysis processes. In addition to this concrete application, theoretical methods are developed for the optimal construction of cluster expansion models, the exact determination of ground states in a large model, and the exhaustive determination of all possible ground states in a small model.

Lastly, a method for nearly-optimal sampling of orientations is presented. Whilst this has many applications in science and engineering, the use-case described here is the indexing of diffraction patterns for experimental materials characterization. Significantly improved sampling is achieved by applying methods from computational geometry.

Resumé

Denne afhandling præsenterer en omformulering af eksisterende problemer inden for materialevidenskab med brug af velkendte metoder fra anvendt matematik: grafteori, numerisk geometri, og heltalsprogrammering.

Centrosymmetriparameteren omformuleres som et grafparringsproblem, hvormed uoverensstemmelserne i de eksisterende beregningsmetoder løses.

Ved at omformulere en afstandsfunktion mellem forskellige krystalgittre (root mean square distance, RMSD) som et parringsproblem i todelte grafer, bevises det, at RMSD i krystalgitre kan beregnes i polynomisk tid, hvilket udgør en forbedring ift. den eksisterende forventning om fakultets-tidsforbrug. Denne metode er derefter udvidet til todimensionelle monolag.

En metode præsenteres til genkendelse af ordnede krystallinske faser i molekuledynamiske simuleringer. En robust klassifikation opnås ved brug af template matching, som også formuleres som et parringsproblem i todelte geometriske grafer.

Matrix dekomponeringer bruges til at udvikle en geometrisk gitterparringsalgoritme, der kan identificere alle grænseflader med lav tøjning. De resulterende stabile og lavenergiske grænseflader som algoritmen finder kan bruges til design og konstruktion af topologiske superledere, som har mange anvendelsesmuligheder inden for kvantecomputere.

Cluster expansion modeller bruges til at finde grundtilstande i guld-sølv nanopartikler, som bruges i en bred vifte af katalytiske processer. Udover denne konkrete anvendelse, udvikles teoretiske metoder optimal konstruktion af cluster expansion modeller, præcis afgørelse af grundtilstanden i større modeller, samt udtømmende bestemmelse af alle mulige grundtilstande i mindre modeller.

Til sidst præsenteres en metode til en næsten-optimal sampling af orienteringer. Blandt de mange anvendelsesmuligheder inden for natur- og ingeniørvidenskab, beskrives her specifikt indeksering af diffraktionsmønstre til brug for eksperimentel materialekarakterisering. Ved at anvende metoder fra numerisk geometri opnås en markant forbedret sampling.

Contents

1 Introduction

Throughout most of human history, progress in materials science has depended exclusively upon physical experimentation and observation of the results. This approach brought such innovations as Damascus steel, Roman concrete, and Chinese porcelain. Whilst these successes were phenomenal achievements based on the technology of the time, the advances were hampered by the slow pace of trial and error experimentation and the inability to observe materials below the macroscale. Building upon the 19th century developments of the atomic theory of materials, electromagnetism, and statistical thermodynamics, the 20th century saw two innovations which changed the way materials science is done: powerful microscopes and the computer.

Non-optical microscopes permit observations of materials at the atomic scale. No longer are observations of the effects of experimentation restricted to the macroscale, but the very physical phenomena which give rise to desirable material properties can be observed directly. Computers have changed the nature of experimentation altogether. Rather than testing new materials in the laboratory, promising materials can be identified by theoretical calculations. Whereas laboratory testing requires many hours or days for each material combination, a computer can screen thousands of materials, in a shorter timeframe and at signicantly lower cost.

Nonetheless, effective materials research requires both experimental and theoretical techniques. A synergistic feedback loop has developed where experimental measurements are used as input to computer simulations, which provide greater understanding of the physical phenomena, and can in turn be used to propose new materials for investigation by experimentalists. Although the methods described here are in some cases not far from a concrete experimental application, this thesis sits firmly on the theoretical side.

1.1 Motivation

Whilst computer studies of materials date back more than half a century, the field has exploded in the last twenty years. The availability of cheap yet powerful commodity hardware has made computer investigations available to everyone, rather than a select few with access to supercomputing facilities. Furthermore, increases in computing power, which has enabled the use of high-level programming programming languages, and the ever-ongoing development of powerful software libraries, have significantly reduced the amount of specialist knowledge required to perform a calculation. As a consequence, data is being produced in greater quantities than ever before. In order to draw meaningful conclusions and gain physical insight from this data, new analysis methods are needed.

In many cases, a lot of effort can be saved by reading the applied mathematical and/or algorithmic literature. For example, microscopists have long since recognized the need to apply methods from computational image analysis, to the extent that this is

now a cornerstone of microscopy. Another, more recent example, is the use of machine learning methods to analyze databases of physical properties, and automatically build predictive models based on these. The ability to recognize a problem in a materials science domain as a classic mathematical problem, and reformulate it as such, has the immediate advantage of being able to leverage the vast literature and man-hours (or man-centuries) invested by the mathematical community.

In this thesis, I present a selection of problems which either permit an elegant reformulation in mathematical terms, or where tools from applied mathematics simplify calculation. In particular, I make use of graph theory, discrete and computational geometry, and mixed integer programming.

Graph theory is well known to materials modellers: it permits a natural representation of a material where nodes and edges correspond to atoms and chemical bonds. Despite this, the breadth and power of graph theory is underappreciated, as we hope to demonstrate here. Discrete geometry is also well known to materials scientists, though typically under different names. For example, the principle of convexity appears in the Wigner-Seitz cell, the Wulff construction of nanoparticles, and forms the basis of many types of optimization. Mixed Integer Programming is a standard tool for practitioners of operations research. It is widely used in industrial planning applications, for everything from nurse roster scheduling, package delivery vehicle routing, shipping containing loading schedules, to train timetabling. Despite this, I can find no evidence of it being used in materials science.

1.2 Outline

The rest of this thesis is ordered as follows: Sections 2-4 describe some selected methods from applied mathematics with are used throughout the thesis. Sections 5-9 present the applications of these. In some cases new and unpublished results are presented; in others, the results of published papers are reframed in the context of the thesis. Section 10 presents an application of the cluster expansion method, which forms the basis of two of the papers. Lastly, Section 11 describes some applications of computational geometry to the indexing of experimental diffraction patterns. Table 1.1 shows the methods used in the work presented here.

Method	Centrosymmetry	Lattice metrics	PTM	Interface matching	AgAu nanoalloy design	Nanoparticle morphology	Alloy design	Pattern indexing
Graph matching	✓	✓	✓	-	-	-	-	-
Canonical forms	-	-	✓	✓	✓	✓	-	-
Point registration	-	✓	✓	✓	-	-	-	-
Convex geometry	-	-	✓	-	-	-	✓	✓
Mixed Integer Programming	-	-	-	-	✓	✓	✓	-
Cluster Expansions	-	-	-	-	✓	✓	✓	-
Quaternion algebra	-	-	✓	-	-	-	-	✓

Table 1.1. Methods used in this thesis. The centrosymmetry parameter is described in section 5. Metrics for lattices are presented in Sections 6 and 7. Polyhedral Template Matching (PTM) is presented in section 8 and published in Paper 1. Interface matching is presented in Section 9 and Paper 2. The cluster expansion method is described in section 10 and forms the basis of the theoretical studies of nanoparticles in Papers 3, 4, and 5. Diffraction pattern indexing is presented in section 11 and Papers 6, 7, and 8.

2 Basic Properties and Operations on Graphs

Many of the methods presented in this thesis are reformulations of a physical problem in graph theoretical terms. Since some aspects of graph theory are not a well-studied topic by all materials science practitioners we will briefly describe some relevant parts here.

2.1 Preliminaries

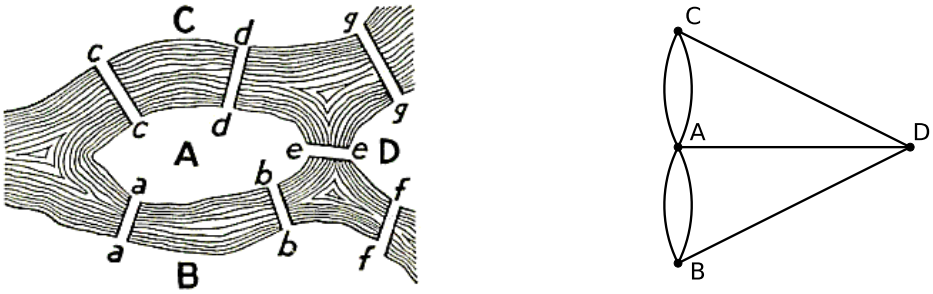


Figure 2.1. Left The seven bridges of Königsberg (image due to Kraitchik [1]). Does there exist a *Eulerian path* through the city: a path which crosses each bridge exactly once?. **Right** A topological representation of the bridges. No *Eulerian path* exists, since the number of vertices of odd degree is neither zero nor two.

Graph theory has its origins in an early 18^{th} century puzzle. The different quarters of the city of Königsberg are separated by the river Pregel, which, at the time, were connected by seven bridges (shown in Figure 2.1). The question was asked: does there exist a route through the city which cross each bridge exactly once? Euler realized that the exact geometry of the bridges was not relevant to the problem. By considering only the connectivity that the bridges represent, he proved that no such walk exists. This insight, that some problems can be reduced to purely topological considerations, forms the foundations of graph theory.

A graph $G = (V, E)$ consists of a set of vertices (or nodes) V and a set of edges E. The edges, $E \subseteq V \times V$, describe the connectivity between vertices, i.e. two vertices a and b are connected if there exists an edge $(a, b) \in E$. As such, a graph is a topological construct which describes relationships between vertices but not the positions of the vertices.

A graph can be either *directed* or *undirected*; in a directed graph, each edge has an orientation such that presence of an edge $(a, b) \in E$ does not imply that $(b, a) \in E$. An undirected graph can be considered a specific case of a directed graph whose connectivity is symmetric. In a *weighted graph*, each edge is assigned a weight. If all edge weights represent a metric function of pairs of vertices, the graph is called a *geometric graph*. Since we are using graph theory to represent physical problems, we will mostly concern ourselves with undirected geometric graphs.

2.2 Isomorphisms, Automorphisms, and Matchings

Whilst there are many details in the methods we present in later sections, they ultimately rely on two properties og graphs; isomorphism and automorphism, and on two variants of a graph matching problem; weighted matching in general and bipartite graphs.

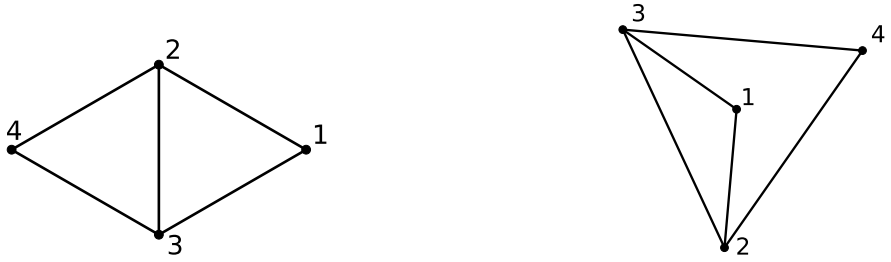


Figure 2.2. Two graphs with 4 vertices and 5 edges. Despite their different vertex labels and different representations as drawings in the plane, the graphs are *isomorphic*. There exists a mapping of the vertices of one graph to the other which preserves the structure of the edge connectivity. A mapping of a graph onto itself it an *automorphism*. The graphs shown here have 4 automorphisms.

Two graphs are called *isomorphic* (lit. the 'same shape') if the vertices of the first graph can be mapped onto the second graph in a way which preserves the vertex connectivity; it is the equality operator on graphs. More formally, two graphs $G = (V, E)$ and $H = (W, F)$ are isomorphic if there exists a bijective function $f : V \to W$ such that for all $v, w \in V$: $\{v, w\} \in E \Leftrightarrow \{f(v), f(w)\} \in F$. An *automorphism* is mapping of a graph onto itself. These concepts are illustrated in Figure 2.2.

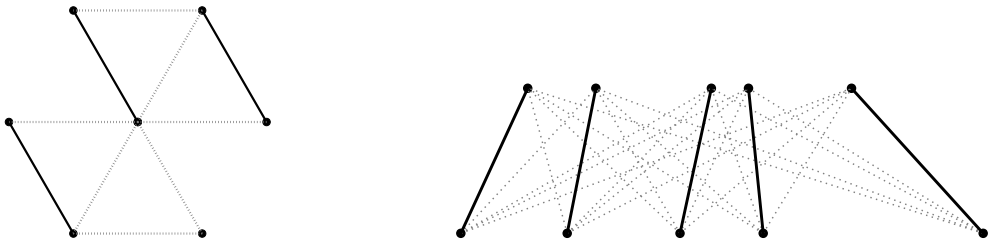


Figure 2.3. Left A maximum-cardinality graph matching in a general graph. No two edges in the matching share a common vertex. **Right** A minimum-weight matching in a bipartite graph. Here, edges exist between every vertex in the top row to every edge in the bottom row. In this graph, the edge weight between two vertices is given by the absolute difference in the x-ordinate.

Given a graph, a *matching* is a subset of edges $\{(a, b) \subseteq E\}$ such that no edges share a common vertex. A *maximum-cardinality* matching is the largest subset of edges which fulfils the matching criterion. In a weighted graph, a *maximum-weight matching* is a matching which maximizes the edge-weight sum. A specific instance of the graph matching problem is on *bipartite* graphs. A bipartite graph has the property that the vertices can be partitioned into two sets, A and B, such that all edges have one vertex

in A and one in B, that is $E = \{(a, b) \in K \mid K \subseteq A \times B, A \cup B = V\}$. Both graph matching variants are shown in Figure 2.3.

Matching problems are applied in diverse settings such as the stable marriage problem [2], optimal identification of donor-recipient pairs for kidney transplantation [3], and the assignment of workers to jobs [4], simply known as the assignment problem. For our applications, however, the graph vertices simply represent atoms, and the edge weights represent a function of the atomic coordinates. Both matching problem variants can be solved efficiently; the general case in $O\left(n^2 m\right)$ using Edmonds' blossom algorithm [5] and the bipartite case in $O\left(n^3\right)$ using the *Hungarian* algorithm [4]. Both algorithms have subsequently been improved upon and have many faster variants, though these come at the cost of increased conceptual complexity.

2.3 Canonical Forms for Isomorphism Testing

We have stated that two graphs with the same vertex connectivity are called isomorphic, but not how to test for isomorphism. A basic method for testing the isomorphism of two graphs is to try every permutation of the vertex labels of one graph, and check if any of the resulting adjacency matrices are equal to that of the other graph. If the isomorphism of many graphs is to be tested, this is an inefficient procedure, since the set of all vertex label permutations must be tested for every pair of graphs. The speed of isomorphism testing can instead be significantly increased by reducing each graph to its canonical form.

Given two graphs $G = (V, E)$ and $H = (W, F)$, a canonical form $C(G)$ is representation such that $C(G) = C(H)$ if and only if G and H are isomorphic. The simplest canonical form is the least element under the lexicographic ordering of all matrices of relevant graph properties. For our applications, the relevant properties are the vertex colours (a optional property used where necessary to distinguish between different vertex types), c_i, and the weight w_{ij} of an edge between two nodes i and j. The property matrix is then given by:

$$\begin{array}{|cccccc|}
\hline
c_{P(1)} & w_{P(1),P(1)} & w_{P(1),P(2)} & w_{P(1),P(3)} & \cdots & w_{P(1),P(n)} \\
c_{P(2)} & w_{P(2),P(1)} & w_{P(2),P(2)} & w_{P(2),P(3)} & \cdots & w_{P(2),P(n)} \\
c_{P(3)} & w_{P(3),P(1)} & w_{P(3),P(2)} & w_{P(3),P(3)} & \cdots & w_{P(3),P(n)} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
c_{P(n)} & w_{P(n),P(1)} & w_{P(n),P(2)} & w_{P(n),P(3)} & \cdots & w_{P(n),P(n)} \\
\hline
\end{array}$$

The canonical form is given by the permutation, P, which minimizes the lexicographic ordering (the dictionary ordering) of the matrix. In practice, this canonical form is inefficient for all but the smallest of graphs, due to the search over all permutations. Practical methods include the widely used `Nauty` [6] library (for general graphs) and Weinberg's method [7] (for polyhedral graphs).

Canonical forms are used in Paper 1 for template matching, where the edge weights are 0 or 1 and simply denote the presence or absence of an edge. In section 10 we use canonical forms to identify equivalent clusters. Here, the vertex colours denote the chemical element of each atom, and the edge weights represent interatomic distances. A geometric canonical form of vectors is key to the performance of the method presented in Paper 2.

3 Point Set Registration

The next fundamental method we will describe is the registration of two point sets. By registration, we mean aligning the points under a set of allowed operations, such that a distance function of the two points is minimized. Typically, the set of allowed operations are rotation, scaling, mirroring, and straining, all of which are linear transformations, and translation. We will first describe rigid point set registration, under which only translation and rotation are permitted, then affine registration, which is effectively a combination of all the linear transformations and translation. Lastly, we discuss methods for calculating strains and illustrate hitherto unnoticed relationships between the methods for calculating strains and the methods for rigid registration.

3.1 Rigid Registration

Given two sets of n points $\mathbf{P}$ and $\mathbf{Q}$, a matching $\mathbf{M} \subseteq \mathbf{P} \times \mathbf{Q}$ is a set of n pairs of points so that each point of $\mathbf{P}$ or $\mathbf{Q}$ appears in exactly one pair. For rigid registration, we wish to find a translation and a rotation of $\mathbf{P}$ such that the distance between the pairs is minimized. A natural distance measure in this case is the Root-Mean-Square Distance (RMSD), defined as:

$$\text{RMSD}\left(\mathbf{M}, \vec{t}\right) = \arg\min_{\mathbf{R},\vec{t}} \sqrt{\frac{1}{n} \sum_{\vec{p},\vec{q} \in \mathbf{M}} \left|\left|\mathbf{R}\vec{p} + \vec{t} - \vec{q}\right|\right|^2} \tag{3.1}$$

where $\mathbf{R}$ and $\vec{t}$ are respectively a rotation and a translation of $\mathbf{P}$. For point sets in $\mathbb{R}^d$ it can be shown that the optimal translation is $\bar{\mathbf{Q}} - \bar{\mathbf{P}}$, where $\bar{\mathbf{P}}$ denotes the barycentre of $\mathbf{P}$. As such, we will dispense with $\vec{t}$ by assuming, without loss of generality, that the barycentres of both point sets lie at the origin.

Figure 3.1. Illustration of rigid registration in $\mathbb{R}^2$. A hexagonal template (blue dots) is registered against a distorted hexagonal structure (orange crosses). **(Left)** The distorted structure is the hexagonal template but with small random perturbations of the vertices. A rotation can be found which minimizes the distance, but no perfect match can be found. **(Right)** Here, the distorted structure is a strained copy of the template. Since the allowed operations in rigid registration are restricted to rotation and translation, a perfect mapping cannot be achieved.

Figure 3.1 demonstrates a simple case of rigid registration using hexagonal templates. In both examples the registration is not perfect, due to the transformations permitted being restricted to rotation and translation only. Whilst the registration is worse than could be achieved by extending the number of allowed transformations, this should not necessarily be considered a drawback. For example, in a crystalline material, atomic displacements of approximately 10% will cause the sample to melt. In a molecular dynamics simulation of such a material, large atomic displacements are indicative of local disorder and should not be accommodated by a template matching process. Thus, depending on the application, rigid registration can be much more robust than non-rigid registration.

3.2 Affine Registration

A generalization of rigid registration is non-rigid, or affine, registration. The set of allowed operations is now extended to include all linear transformations and translation:

$$\mathrm{RMSD}\left(\mathbf{M}, \vec{t}\right) = \underset{\mathbf{A},\vec{t}}{\arg\min} \sqrt{\frac{1}{n} \sum_{\vec{p},\vec{q}\in\mathbf{M}} \left|\left|\mathbf{A}\vec{p} + \vec{t} - \vec{q}\right|\right|^2} \tag{3.2}$$

where $\mathbf{A}$ is a linear map. Unlike a rotation matrix, which is constrained to be right-handed and orthonormal, a linear map is unconstrained (i.e. an arbitrary matrix). As in the rigid case, the optimal translation is $\bar{\mathbf{Q}} - \bar{\mathbf{P}}$, and we again dispense with $\vec{t}$ by assuming that the barycentres of both sets lie at the origin.

The optimal linear map can be determined easily. If $n = d$, $\mathbf{A}$ can be determined by solving a system of linear equations. If $n \geq d$ the system is overdetermined and is solved using a least squares fit. If used in continuum mechanics context, the linear map $\mathbf{A}$ is known as the elastic deformation gradient. In the overdetermined case, the residual term of the least-squares fit is used for identifying local irreversible shear transformations [8].

The elastic deformation gradient is arguably most intuitive when used to calculate strains. To do so, we use a polar decomposition, which splits a deformation gradient into a rotational (and possibly a mirroring) component, and a strain component. The polar decomposition is well known to matrix mathematicians, and computer graphics practitioners. In a physical context, though, it appears to be a hidden treasure of continuum mechanics which is little known outside of this field.

The polar decomposition of a square matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ is written:

$$\mathbf{A} = \mathbf{ZH} \tag{3.3}$$

where $\mathbf{Z}$ is an orthogonal matrix and $\mathbf{H}$ is a positive semidefinite Hermitian matrix (since we do not consider imaginary numbers, $\mathbf{H}$ is simply symmetric). This is quite a remarkable result; it implies that a deformation gradient can always be expressed as a symmetric strain matrix, given that the system is viewed in the appropriate coordinate system. As such, the strain between two point sets can always be calculated, regardless of their relative orientation or chirality.

Figure 3.2. Illustration of non-rigid registration in $\mathbb{R}^2$. A hexagonal template (blue dots) is registered against the same distorted hexagonal structure (orange crosses) as those in Figure 3.1. **(Left)** Affine registration permits a significantly better registration against the randomly perturbed vertices than rigid registration, though a non-linear distortion is still present. Affine registration cannot accommodate non-linear distortions. **(Right)** The alignment of the template onto the strained copy is a perfect match.

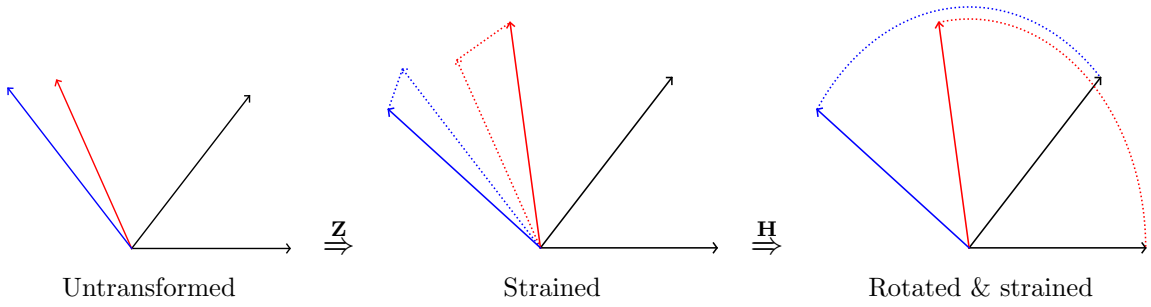


Figure 3.3. Illustration of the polar decomposition in $\mathbb{R}^2$. The first vector pair (coloured red and blue) can be mapped onto the second pair (black) with a linear transformation (the deformation gradient). This transformation can be decomposed into an orthogonal matrix (in this case a rotation) and a symmetric matrix (a strain). The untransformed vectors (**left**) are first strained (**centre**) and then rotated (**right**) onto the second pair of vectors. Since the number of points to register is equal to the dimension of the Euclidean space, the registration is exact.

Figure 3.3 demonstrates the polar decomposition of the linear mapping of one pair of vectors onto another. The transformations are, in order, a strain and a rotation. This follows the order in equation (3.3), which defines a *right-handed* polar decomposition. Depending on the application, a left-handed decomposition, $\mathbf{A} = \mathbf{HZ}$, may be preferred instead. Whilst the net result is the same, the strain matrices are not in general identical, due to the non-commutativity of the matrix product.

3.3 On the Relationship of Optimal Rotation Matrices to the Polar Decomposition

In two dimensions, the polar decomposition and the optimal rotation matrix for rigid registration permit trivial derivations. In higher dimensions these are both non-trivial. In any number of dimensions, though, there exists an intimate connection between the two calculation methods which appears to have gone unnoticed in the literature.

Of the two calculations, the polar decomposition is the oldest; Horn [9] provides a good overview of its early history. Notable developments include its invention by Autonne [10], subsequent rediscovery by Wintner and Murnaghan [11], development of a practical Singular Value Decomposition (SVD) algorithm by Golub and Kahan [12] which also enables a polar decomposition, a Newtonian algorithm with quadratic convergence by Higham [13], and, of particular relevance for material modelling, the recent development of a fast polar decomposition for 3×3 matrices, by Higham and Noferini [14].

Calculation of the optimal rotation matrix also has an interesting history, which Karney [15] describes in detail. The method has been independently discovered at least four times, each of which has produced separate developments, including in spacecraft attitude determination [16, 17, 18], robotics [19], analysis of aerial photograph geometry [20], and structural biology [21, 22, 23].

The fiducial method for both calculations in any number of dimensions uses the SVD. A real-valued matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ can be decomposed:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{3.4}$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V}^T \in \mathbb{R}^{n \times n}$ are unitary matrices, and $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is a non-negative diagonal matrix. Then, the factors of the polar decomposition of $\mathbf{A} = \mathbf{ZH}$ are given by $\mathbf{H} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{Z} = \mathbf{U}\mathbf{V}^T$.

The calculation of the optimal rotation matrix using the SVD is due to Kabsch [24]. We let $\mathbf{M}$ be the cross-covariance matrix $\mathbf{A} = \mathbf{P}^T\mathbf{Q}$, where $\mathbf{P}$ and $\mathbf{Q}$ are the matrix notation representations of the point sets. Applying the same decomposition as in Equation (3.4), the optimal rotation matrix is given by:

$$\mathbf{R} = \mathbf{V} \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & & \dots & \\ 0 & 0 & 0 & \dots & \kappa \end{bmatrix} \mathbf{U}^T \tag{3.5}$$

where $\kappa = \det\left(\mathbf{V}\mathbf{U}^T\right)$ is a correction which ensures that an improper rotation is avoided. The relationship between the two calculations is already obvious: other than a formulation-specific transposition, the difference is simply in the requirement for a right-handed orthogonal matrix, i.e. a proper rotation with a determinant of 1.

The connection between the two becomes even clearer when considering recent algorithms used for the calculation an optimal rotation matrix in $\mathbb{R}^3$ and for the polar decomposition of a 3×3 matrix, both based on quaternions. Quaternions are unit vectors in $\mathbb{S}^3$, which is a double covering of $SO(3)$ [25], and can therefore only express proper rotations. Theobald [26] adapts the method of Horn [20] to find a rotation which satisfies:

$$\mathbf{R} = \underset{\mathbf{X} \in SO(3)}{\arg\max}\, \mathrm{tr}\left(\mathbf{X}\mathbf{A}\right) \tag{3.6}$$

Here, $SO(3)$ denotes the space of orientations. Higham and Noferini [14] show that the optimal orthogonal matrix is determined by:

$$\mathbf{A} = (\eta \mathbf{Z}) \cdot \mathbf{H} \Longleftrightarrow \underset{\mathbf{X} \in SO(3)}{\arg\max}\, \eta \operatorname{tr}\left(\mathbf{X}^T \mathbf{A}\right) \tag{3.7}$$

where

$$\eta = \operatorname{sign}(\det \mathbf{A}) = \begin{cases} 1, & \text{if } \det \mathbf{A} \geq 0 \\ -1, & \text{if } \det \mathbf{A} < 0 \end{cases} \tag{3.8}$$

By restricting the search to $SO(3)$ through the use of quaternions, Equation (3.3) avoids the correction needed in Equation (3.5). Equation (3.7) does the inverse, restricting the search to $SO(3)$ and then determining if a correction is needed.

Whilst these methods are clearly intimately related, this observation does not appear to be present in the literature. The crystallographic community has noted the relationship between improper rotations and enantiomorphs [27], but not the connection to the polar decomposition. The two methods have developed separately, in parallel. No cross-citation between articles from the two communities could be found either. A possible explanation for this is that the RMSD is typically only calculated in $\mathbb{R}^2$ or $\mathbb{R}^3$, whereas the polar decomposition is used in any number of dimensions.

A practical consequence of this relationship is that any polar decomposition method can be adapted to calculate optimal rotation matrices, and vice versa. We have done exactly this, by adapting the optimal rotation matrix method of Theobald [26] and Liu et al. [23] to create the fastest polar decomposition library for 3×3 matrices†.

3.4 Applications in this Work

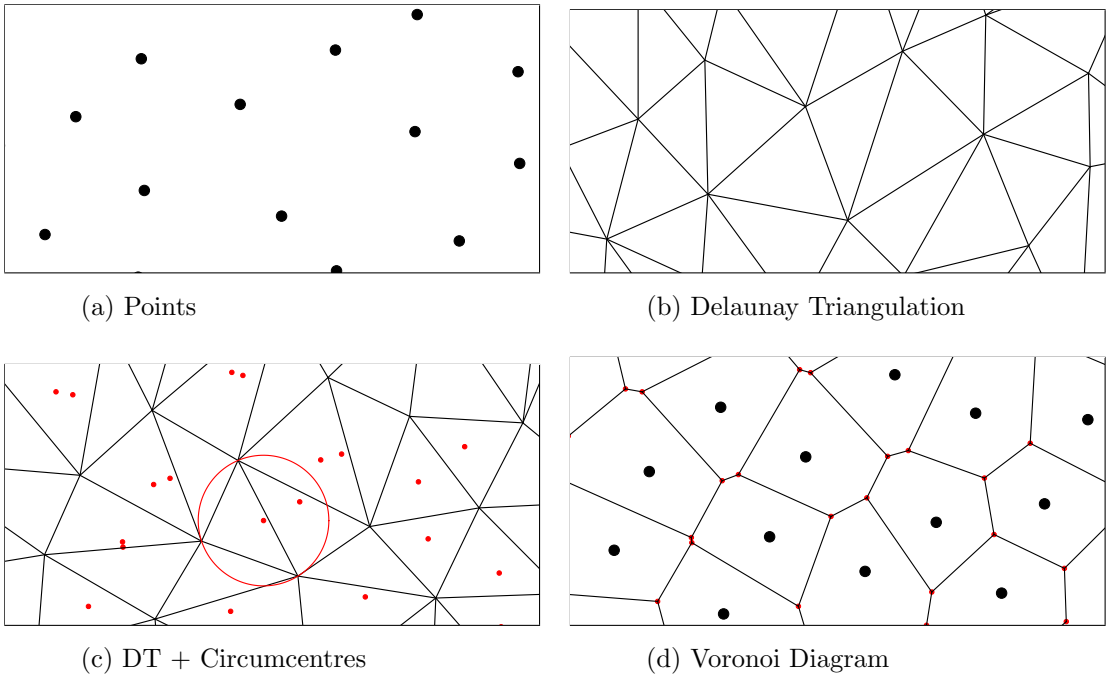
Both equations (3.1) and (3.2) require a matching, i.e. a point-to-point correspondence between the sets, but do not state how this should be obtained. Two sets of n points can be registered in $n!$ ways, which is infeasible for all but the smallest of sets. Since the RMSD in both equations (3.1) and (3.2) are nonlinear functions of respectively rotation and linear mapping, the standard algorithm for bipartite matching (c.f. subsection 2.2) cannot be used. Determination of the optimal matching are the central innovations of Paper 1 and the crystal metric described in section 6. Both of these methods use the RMSD as a distance function. Non-rigid registration and the polar decomposition form the basis of the interface matching method presented in Paper 2.

† Higham and Noferini's [14] algorithm is possibly faster, but, at the time of writing, exists only as MATLAB code [28]. For a fair comparison it should be rewritten in a faster language such as C. They also provide a rigorous analysis of numerical stability for their method, something which Theobald's method lacks. I thank the authors for sharing their code with me before it was made publicly available.

4 A Selection of Computational Geometry Tools

Many of the applications in this thesis make extensive use of methods from computational geometry. In particular, we use the convex hull, Delaunay triangulation, and the Voronoi diagram. For a detailed introduction we refer to standard textbooks [29, 30], but here we describe the relevant concepts in brief.

4.1 The Holy Trinity



(a) Points

(b) Delaunay Triangulation

(c) DT + Circumcentres

(d) Voronoi Diagram

Figure 4.1. A set of points in the plane (a), its Delaunay triangulation (b) with circumcentres and a single circumcircle shown (c), and the Voronoi diagram (d).

Figure 4.1 shows each of the methods in action on the same set of points (a). A triangulation of a point set is a partition into non-overlapping simplices (b). In $\mathbb{R}^2$, a simplex is simply a triangle, but in the general case a simplex in $\mathbb{R}^d$ has $d + 1$ vertices. The vertices of a simplex define a hypersphere, which in $\mathbb{R}^2$ is simply a circle. This hypersphere is called the circumhypersphere, and the midpoint is called the circumcentre. The *Delaunay* triangulation [31] has stricter requirements than a plain triangulation: it must fulfil the empty-sphere condition. This means that no point in the triangulation may lie within the circumhypersphere of another simplex. A single (empty) circumcircle is shown in (c). The Voronoi diagram (d) is a partitioning of the space into contiguous convex regions, known as Voronoi cells [32]. The Voronoi cell of each point is the region of space which lies closer to the point that to any other. The extent of a cell is defined by its extreme vertices, which are known as Voronoi vertices. Here, the relationship between the Delaunay triangulation and the Voronoi diagram becomes clear: each circumcentre in the Delaunay triangulation is a Voronoi

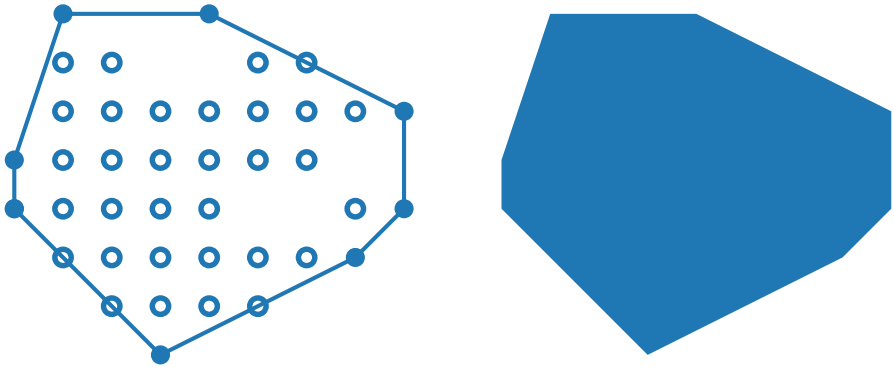


Figure 4.2. Illustration of the convex hull and the maximum principle. Given a set of points, the convex hull is the smallest convex set which contains all of the points. **Left:** A set of points in $\mathbb{R}^2$ and the planes which define extent of the convex hull. Extreme vertices are marked by closed circles, interior vertices are marked by open points. Although the points shown here have integral coordinates, the principle is general and applies to any set of points in any number of dimensions. **Right:** The convex hull of the points is a compact (closed and bounded) convex set. By the maximum principle, any linear function of a compact convex set achieves its maximum at the boundary of the set.

vertex. For this reason, the Delaunay triangulation and the Voronoi diagram are the dual representations of each other.

The convex hull [33] of a set of points is the smallest convex set which contains all the points. The boundary of a convex hull can be defined by a set of hyperplanes, which in $\mathbb{R}^2$ are lines. In two dimensions, an oft-used analogy is that the boundary of the convex hull has the same shape as an elastic band placed around the points. As well as its geometric significance, the convex hull has significance in linear algebra and optimization. The *maximum principle* [34] states that any linear function of a compact convex set attains its maximum on the boundary of the set. These concepts are illustrated in Figure 4.2.

Whilst the convex hull is usually applied in Euclidean spaces, it can also be used to calculate Delaunay triangulations in curved spaces. Consider a set of points on a sphere; if we calculate a standard Delaunay triangulation, the resulting simplices will traverse the sphere interior. A triangulation of the sphere *surface* is called a spherically constrained Delaunay triangulation. By calculating a Euclidean convex hull, the simplices of the spherically constrained Delaunay triangulation are found. As illustrated in Figure 4.3, the simplices in both spaces contain the same vertices.

We use these methods extensively throughout this thesis. The Voronoi diagram and the convex hull are applied in Paper 1, also described in section 8. The convex hull and the maximum principle form the basis of Paper 5. The spherically constrained Delaunay triangulation, calculated using the aforementioned convex hull trick, permits the accurate measurement and optimization of hyperspherical coverings, described in Paper 6.

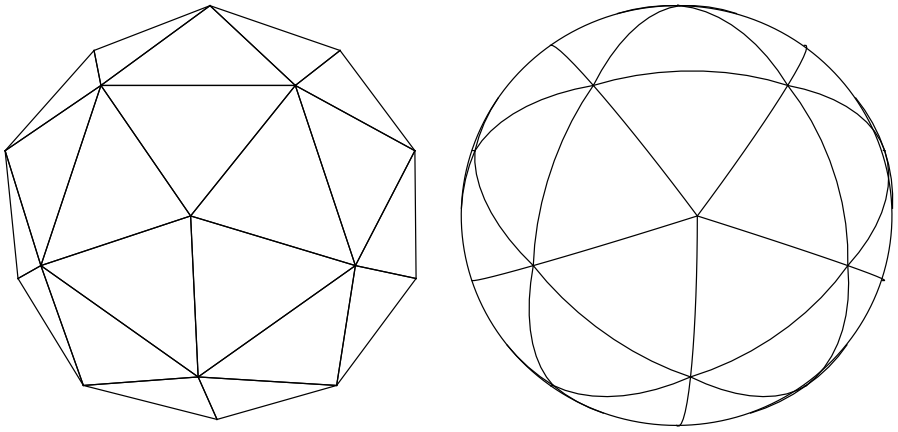


Figure 4.3. Convex hull (left) and the spherically constrained Delaunay triangulation (right) of 22 points on the sphere. The triangulations exist in $\mathbb{R}^3$ and $\mathbb{S}^2$ respectively, but the vertices of each simplex are the same. Data due to Sloane [35]. Figure reproduced from Paper 6.

5 Correct Formulation of the Centrosymmetry Parameter

Here, we provide a first demonstration of the usefulness of reformulating a problem in graph theoretical terms. The centrosymmetry parameter [37] is widely used for analysis of condensed phases in molecular dynamics simulations, as illustrated in Figure 5.1. Despite the paucity of information it provides compared to more modern methods [38, 39, 40], it is still commonly used for identification of ordered lattice phases. A recent development is its use in the identification and classification of different types of defect structures [41, 42]. These structures are often highly acentrosymmetric, which, as we will show, causes the existing CSP calculation methods to fail. In this chaper, we first introduce the original formulation of the CSP for FCC lattices. We then reformulate the problem as a graph matching problem, and show that the existing methods for calculation of the CSP do not meet this description. The disparity between the correct and existing methods is demonstrated using data from large-scale simulations. Lastly, we discuss the implementation of the correct method.

5.1 Centrosymmetry Parameter

The centrosymmetry parameter was originally defined by Kelchner et al. as:

$$p = \sum_{i=1}^{6} |\vec{r}_i + \vec{r}_{i+6}|^2 \tag{5.1}$$

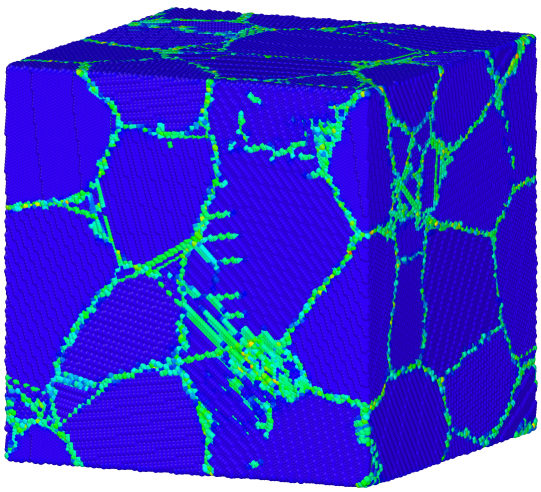


Figure 5.1. Centrosymmetry analysis of a simulated Cu_3Pt nanocrystalline sample. The CSP can distinguish between centrosymmetric atoms, such as FCC (blue), and acentrosymmetric atoms such as HCP or disordered atoms (cyan). Analysis and rendering done using OVITO [36]

.

Variables:	$x_{i,j} \in \{0,1\}$	$\forall i,j$	Edges on/off	(5.2)
Parameters:	$w_{i,j} = \lvert\vec{r}_i + \vec{r}_j\rvert^2$	$\forall i,j$	Edge weights	(5.3)
Minimize:	$\frac{1}{2}\sum_{i,j} x_{i,j} w_{i,j}$		Minimize the edge weight sum	(5.4)
Subject to:	$\sum_j x_{i,j} = 1$	$\forall i$	Neighbour row constraint	(5.5)
	$\sum_i x_{i,j} = 1$	$\forall j$	Neighbour column constraint	(5.6)
	$x_{i,i} = 0$	$\forall i$	No self-neighbours	(5.7)
	$x_{i,j} = x_{j,i}$	$\forall i,j$	Neighbour reciprocity	(5.8)

Model 5.1. A mathematical model for exact calculation of the centrosymmetry parameter. Edges between vertices (5.2) are formulated using an adjacency matrix. The matching conditions are enforced by Equations (5.5)-(5.8) This model represents a straightforward reformulation of the CSP in terms of a graph matching problem.

where $\vec{r}_i$ and $\vec{r}_{i+6}$ are the vectors 'corresponding to the six pairs of opposite nearest neighbours in the FCC lattice' [37]. The extension to BCC (as well as other centrosymmetric structures) is intuitive, as each nearest neighbour atom has a clearly defined opposite neighbour. In the general case though, we argue that the CSP should be defined as a minimum-weight maximum-cardinality graph matching.

Given a graph $G = (V, E)$ where V and E are the vertices and edges respectively, a matching M is a subset of E such that no two edges in M share a common vertex. A matching is *maximal* if it is not a subset of any other matching. A matching has *maximum cardinality* if it is at least as large as any other matching of G. In a *perfect matching* all vertices are adjacent to an edge in M. In our case, each vertex represents an atom, and each edge represents a neighbour relationship between two atoms. Since we have an even number of atoms and we allow an atom to pair with any atom other than itself, G is a complete graph, and any maximal matching is therefore also a maximum cardinality matching and a perfect matching. Our task is therefore to find a maximal matching which minimizes the sum of the edge weights.

The problem is stated formally in Model 5.1. Although the CSP was not originally defined for structures other than FCC, the model formulation meets the intuitive requirements: the CSP should have the lowest possible value whilst ensuring that each atom has exactly one opposite neighbour and that opposite neighbours are reciprocal.

There are currently two commonly used methods for calculating the CSP, which we will denote *Greedy Edge Assignment* (GEA) and *Greedy Edge Matching* (GEM). Neither method guarantees that all the conditions of a minimum-cost matching are met. GEA (described by e.g. Stukowski [39] and implemented in LAMMPS [43] and OVITO [36]) finds the $\frac{N}{2}$ lowest edge weights of the $\frac{N(N-1)}{2}$ possible opposite neighbour pairs. This does not guarantee that every atom has exactly one opposite neighbour;

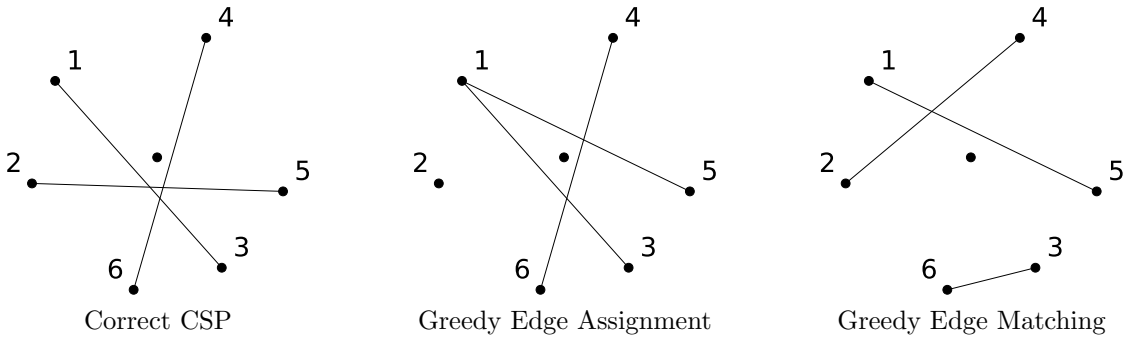


Figure 5.2. A 2D example comparing the correct CSP calculation with the GEA and GEM methods. The vertex labels are ordered by distance from the central atom. In this case, a relatively small deviation from perfect centrosymmetry is sufficient to induce failure in the greedy methods: GEA does not produce a valid matching (some atoms have no opposite neighbours, others have multiple opposite neighbours); GEM produces a valid matching, but not of minimum weight. The CSP values calculated are 0.39, 0.32 and 4.36 respectively.

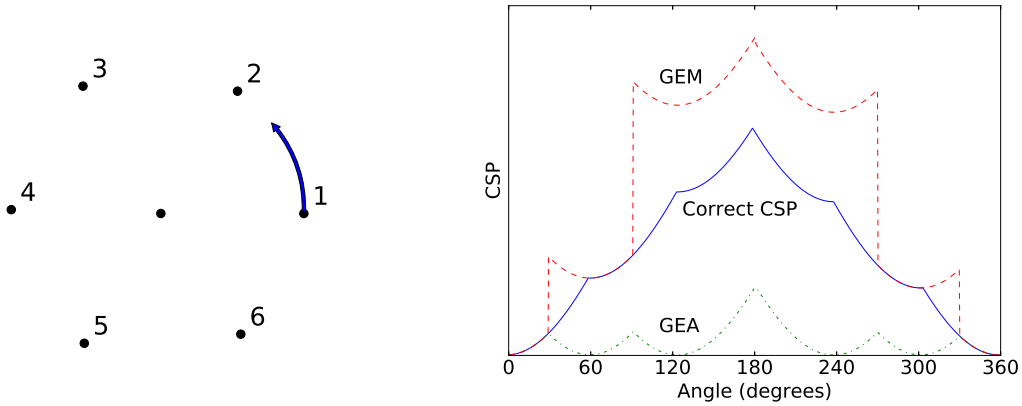


Figure 5.3. **Left** A simple example to illustrate the failure modes of greedy CSP calculation methods. The centrosymmetry of a slightly perturbed hexagonal structure is changed by rotating a single vertex about the central atom, through an angle range of 2π. Vertex labels are ordered by distance from the central atom. **Right** The CSP values calculated using all three methods. GEM is not a continuous function of rotation. GEA consistently underestimates the actual CSP.

the constraints specified by equations (5.5) and (5.6) are effectively relaxed and the calculated CSP is consequently a lower bound on the true CSP. In GEM (descibed by e.g. Bulatov et al. [44] and Li [45] and implemented in AtomEye [46]) the atoms are sorted by distance from the central atom. The opposite neighbour of the innermost atom is chosen by minimizing the edge weight. Both atoms are then removed from the set and the process is repeated until no atoms are left. This method always produces a valid graph matching, but does not guarantee that the edge weight sum is minimal, and is therefore an upper bound on the true CSP. Figure 5.2 shows a two dimensional example on which both GEA and GEM fail.

The failure modes of the greedy methods can be demonstrated with a simple example. Figure 5.3 shows a hexagonal structure with slightly perturbed vertices. The

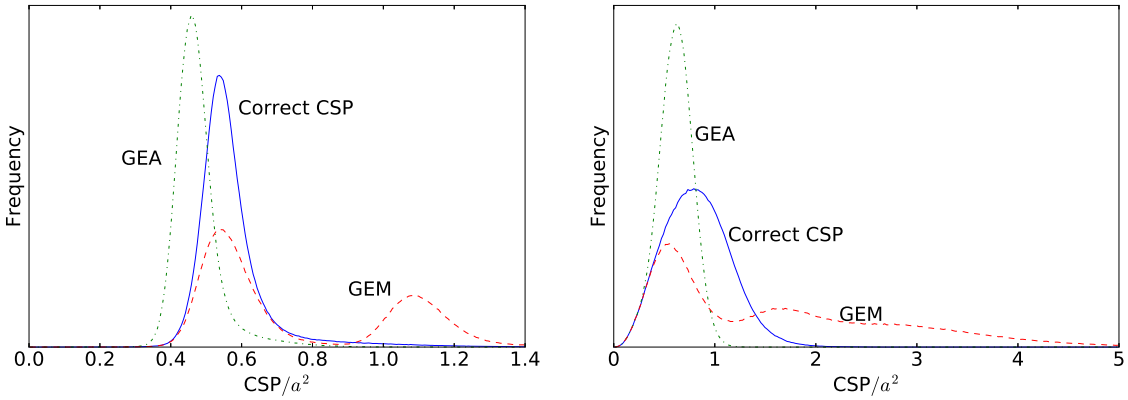


Figure 5.4. Histograms of the correct CSP calculation, Greedy Edge Matching, and Greedy Edge Assignment methods for all HCP atoms in a polycrystalline Ru sample, and all non-FCC atoms in a polycrystalline Cu_3Pt sample. The atoms were selected using the PTM method [40]. The CSP values have been normalized by a pseudo-lattice-constant. In the HCP case, this is the $\sqrt{2}$ times the interatomic distance. The FCC lattice constant is used for the non-FCC atoms.

first atom is rotated through a 2π angle range, and the CSP values for each method is calculated at every rotation. As well as significantly overestimating the CSP in many intervals, GEM is not a continuous function of rotation; small changes in geometry cause large changes in the calculated CSP value. GEA, on the other hand, fails to consistently increase the calculated CSP, even when the centrosymmetry of the structure is clearly higher. Only the graph matching calculation exhibits the expected behaviour.

5.2 Application to Molecular Dynamics Simulation Data

Although the hexagonal example described above highlights the failure modes of the greedy CSP calculation methods, it is clearly a pathological case, since the close proximity of two atoms is unphysical. To illustrate the practical consequences of the use of the greedy methods in a realistic setting, we have calculated the CSP of every atom in two systems: a FCC Cu_3Pt system containing 2.8 million atoms (shown in Figure 5.1), and a HCP Ru system containing 2.5 million atoms.

Figure 5.4 shows histograms of each method. For both the HCP and non-FCC atoms, the different methods give markedly different results. In the aforementioned systems, GEA varies by as much as a factor of 5.5 from the correct CSP; GEM varies by as much as a factor of 13.2. We have not shown a histogram for FCC or BCC atoms, as these structures have low CSP values at low temperatures, and consequently both GEA and GEM perform well in these cases. However, a common current use case of the CSP is in identifying specific defect types, which have high CSP values.

5.3 Implementation

The number of matchings in a complete graph with $n+1$ vertices is given by the double factorial function:

$$n!! = \prod_{k=0}^{\lceil n/2 \rceil} (n - 2k) \tag{5.9}$$

For $n + 1 = 12$ and $n + 1 = 14$ this gives 10,395 and 135,135 possible matchings respectively. Evaluating all possible solutions would incur a significantly performance penalty. Instead, we can use one of the many polynomial-time algorithms for minimum-weight maximum-cardinality matching (primarily variations of Edmonds' $O(n^2m)$ Blossom algorithm [5]; c.f. Kolmogorov's article [47] for a good overview). We have used the LEMON [48] graph library to calculate matchings. On a test of 14 vertices with randomly distributed atom coordinates, we can calculate approximately 50,000 CSPs per second on a single core of a 2014 MacBook Pro with an Intel Core i7-4770HQ 2.20GHz CPU.

A significant performance increase can be achieved with a hybrid strategy. The CSP is first calculated using GEA. If the edge assignment constitutes a valid matching, the CSP is correct and we are done. Otherwise, we use the graph library to calculate the CSP. With this approach, the (slower) graph library method is only invoked when the (fast) greedy method fails.

6 A Metric for Lattices †

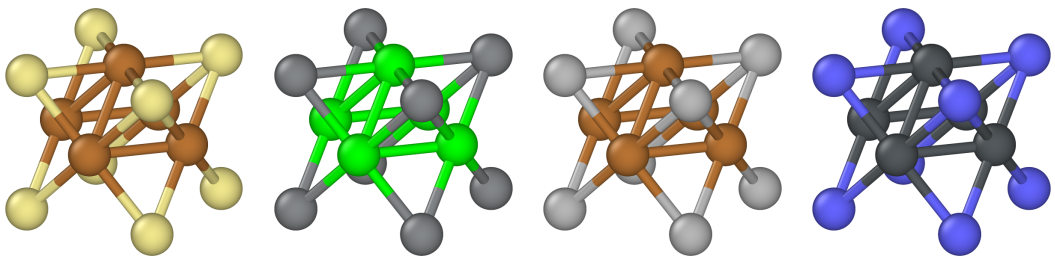
We continue our theme of applying geometric graph theory to the analysis of materials, this time by using bipartite matching to calculate a distance (the RMSD) between lattices. Calculation of the RMSD in nanoparticles, organic molecules and bulk crystalline materials has been conjectured to be computationally infeasible for all but the smallest of systems, due to a supposed factorial scaling of the running time with the number of atoms. For the latter case, we provide a proof that the RMSD can be calculated in polynomial time.

We first describe the motivation for use of the RMSD, then describe an algorithm to compute the RMSD in polynomial time. To illustrate the method, we perform a cluster analysis on a small selection of materials from the Inorganic Crystal Structure Database (ICSD) [49]. In the next section we adapt the algorithm to calculate the RMSD in 2D materials.

6.1 Motivation

The ability to calculate a distance between crystal structures has many applications, both experimental and computational. In an experimental context, the applications of a crystal metric are primarily detection of duplicate crystal structures and the identification of crystallographic prototypes. Crystal structures can be determined experimentally using a range of electron, X-ray or neutron diffraction modalities. Once the lattice parameters and atomic positions have been determined, it is essential to test whether the structure is already known, in order to avoid publication of duplicate structures as well as unnecessary entries in structure databases. If a structure is found to be new, one must search for similar structures in order to assign a crystallographic prototype. For example, the L_{12} decoration of a FCC lattice structure is called the $AuCu_3$ prototype, because this was the first structure of this type to be experimentally

† The work in this section has been done in collaboration with K.W. Jacobsen and J. Schiøtz. We intend to submit it for publication, though at the time of writing this has not yet been done.



$AuCu_3$	$AlNi_3$	$PtCu_3$	$GaFe_3$
$d = 3.77$Å	$d = 3.56$Å	$d = 3.71$Å	$d = 3.66$Å

Figure 6.1. Four structures with the $AuCu_3$ crystallographic prototype. The structures have different chemical compositions and lattice constants, but otherwise have the same overall geometric structure.

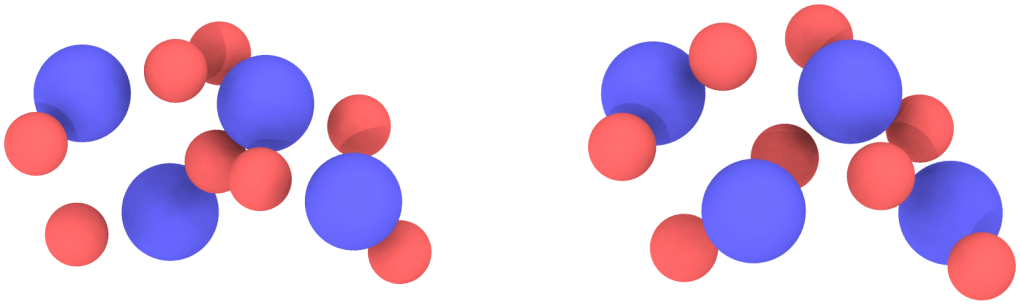


Figure 6.2. Initial (left) and relaxed (right) structure of SnF_2. Relaxation introduces significant changes in the positions of the F atoms. Structures taken from the Materials Project [50].

determined. All structures which have the same overall geometry but at a different length scale, i.e. a difference in lattice constants and chemical elements only, are assigned the same crystallographic prototype. For example, $PtCu_3$, $AlNi_3$ and $GaFe_3$ all have the $AuCu_3$ prototype, as shown in Figure 6.1.

For computational studies a crystal metric also has many different applications. In recent years increased computational resources as well as determined efforts to find new materials computationally have resulted in ever increasing numbers of crystalline structures, which the research community has started to organize systematically in databases such as NOMAD [51], AFLOWLIB [52], and OQMD [53]. As in the experimental case, duplicate detection is required to avoid populating databases with redundant entries.

A further challenge which is unique to computational studies is in determining changes due to structural relaxation. The ab-initio methods used to calculate energies require many hours or even days of computation. When searching for a stable structure, materials are often screened for stability by calculating the energy at a single geometric configuration. Promising structures are then relaxed, whereby the atoms are moved to the positions which (locally) minimizes the structural energy, c.f. Figure 6.2. Determining the magnitude of the structural relaxation, i.e. whether the overall geometry has been preserved or not, is a challenging task and often done by manual inspection.

Another computational use case is in crystal structure prediction using evolutionary techniques. Here, the space of configurations is explored using e.g. genetic algorithms, in order to (hopefully) find the global optimum. Genetic algorithms typically include diversification and intensification stages, which serve the respective purposes of ensuring that a broad range of configurations are explored, and that promising configurations are refined. To ensure diversification, new structures proposed by the genetic algorithm are compared against all previously investigated structures. Proposals which are not sufficiently original are discarded, in order to reduce the number of ab-initio energy calculations. Methods such as USPEX [54] use a *chemical fingerprint* [55] to measure structural similarity, though we conjecture that the RMSD might be a better measure

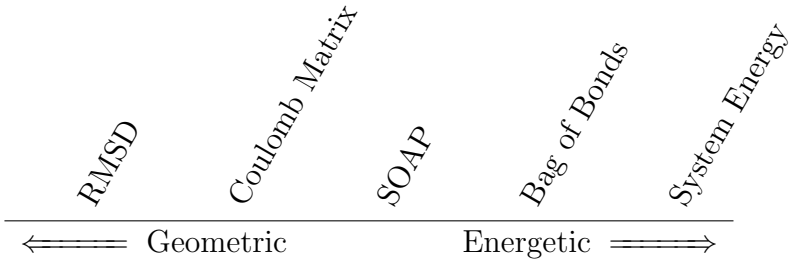


Table 6.1. Qualitative comparison of a selection of methods for measuring crystal similarity. The descriptors are ordered according to the nature of the comparison operation (primarily geometric or primarily energetic). The RMSD, which is purely geometric, and the system energy, which is purely energetic, represent extreme points on this axis.

of distance.

In the literature there exist several methods for detecting duplicate crystal structures and assigning structural prototypes. Efforts to quantify similarity include the use of chemical fingerprint descriptors [55, 56, 57, 58] and duplicate detection using purely geometric methods [59, 60, 53]. The duplicate detection methods share the common traits of requiring a distance threshold to be chosen *a priori*, and do not constitute a metric, as they effectively calculate only two distances, 0 and ∞, corresponding to duplicate and non-duplicate structures respectively.

Whilst some fingerprint distances do constitute proper metrics [57, 61], others have the primary goal of capturing energetic effects, in order that potential energy surfaces can be constructed using these descriptors or in order to build predictive physical models from existing data. We have ordered a selection of metrics on a geometric/energetic scale in Table 6.1. The Coulomb matrix [62] is a function of atomic numbers and reciprocal interatomic distances; this choice of distance reflects the influence of the short-ranged nature of physical interactions upon the system energy. The Smooth Overlap of Atomic Positions (SOAP) [56] descriptor also considers only short-ranged neighbour relationships, but more aggressively so, by using a cutoff. The Bag of Bonds (BoB) model [63] constructs a set of interatomic bonds, after which the geometric information is discarded. This is therefore much more an energetic descriptor than a geometric descriptor.

The RMSD represents the 'most geometric' comparison method on this axis. As a good descriptor should be, it is permutationally and translationally invariant, and it is arguably the most intuitive metric, since it is a simple distance function. It has also widely been regarded as being computationally infeasible due to a postulated factorial running time [57, 58]. We now prove that the RMSD can be calculated in polynomial time.

6.2 Matchings in Crystal Lattices

Given two sets of n points $\mathbf{U}$ and $\mathbf{V}$, both in $\mathbb{R}^d$, a matching $\mathbf{M} \subseteq \mathbf{U} \times \mathbf{V}$ is a set of n pairs of points so that each point of $\mathbf{U}$ or $\mathbf{V}$ appears in exactly one pair. The Root-Mean-Square Deviation (RMSD) between $\mathbf{U}$ and $\mathbf{V}$ is defined as:

$$\mathrm{RMSD}\left(\mathbf{M}, \vec{t}\right) = \sqrt{\frac{1}{n} \sum_{(\vec{u},\vec{v}) \in \mathbf{M}} \left|\left|\vec{u} + \vec{t} - \vec{v}\right|\right|^2} \tag{6.1}$$

where $\vec{t} \in \mathbb{R}^d$ is a translation of $\mathbf{U}$. A crystal lattice consists of a set of basis atoms which are repeated by translation, along all integer multiples of the lattice vectors and infinitely in extent. Due to the infinite repetition in space, two lattices can only be meaningfully compared (geometrically, at least) if their unit cells are identical. Given this constraint is satisfied, the RMSD for a two crystal lattices can be expressed as:

$$\mathrm{RMSD}_{\mathrm{lattice}}(\mathbf{M}, \vec{t}) = \sqrt{\frac{1}{n} \sum_{(\vec{u},\vec{v}) \in \mathbf{M}} \arg\min_{\vec{h}} \left|\left|\vec{u} + \vec{t} - \left(\vec{v} + \mathbf{C}^T \vec{h}\right)\right|\right|^2} \tag{6.2}$$

where $\mathbf{C} \in \mathbb{R}^{d \times d}$ is the lattice vector matrix. Here, $\vec{h} \in \mathbb{Z}^d$ specifies an integer multiple of the lattice vectors, which allows a point in $\mathbf{U}$ to match a point in an adjacent cell in $\mathbf{V}$. This formulation is a complete and consistent description of the RMSD between two crystal lattices; whilst the distance is only calculated over the atoms of a single unit cell, the distances are the same in every other unit cell.

The distance we are looking for is the smallest possible one, given by:

$$d\left(\mathbf{U}, \mathbf{V}\right) = \arg\min_{\mathbf{M}, \vec{t}} \mathrm{RMSD}_{\mathrm{lattice}}\left(\mathbf{M}, \vec{t}\right) \tag{6.3}$$

From the above equation it is clear that there are an infinite number of translations and $n!$ possible matchings; this is the reason for the postulated computational infeasibility of determining the optimal RMSD. We will now prove that only a polynomial number of matching must be investigated.

6.3 Bounding the Number of Minimizing Matchings

In any balanced complete bipartite graph with n nodes there are $n!$ possible matchings. However, the weights in the bipartite graph of our lattice matching problem vary only with lattice translation. Clearly, a matching $\mathbf{M}$ is only of interest if there exists some translation $\vec{t}$ such that the matching has a lower RMSD than all other matchings at $\vec{t}$, that is:

$$\exists \vec{t} \quad s.t. \quad \mathrm{RMSD}_{\mathrm{lattice}}(\mathbf{M}, \vec{t}) \leq \mathrm{RMSD}_{\mathrm{lattice}}(\mathbf{M}', \vec{t}) \quad \forall \mathbf{M}' \tag{6.4}$$

We denote a matching as a *minimizing matching* if this criterion is fulfilled. We give a simple proof that, for a lattice in $\mathbb{R}^d$ with n points in the unit cell, the number of minimizing matchings is at most n^d.

Lemma 1. *Consider a matching of two (non-periodic) point sets* $\mathbf{U}$ *and* $\mathbf{V}$ *and a translation* $\vec{t}$ *of* $\mathbf{U}$*. The optimal matching is independent of* $\vec{t}$*.*

Proof. A rearrangement of equation (6.1) gives:

$$n \cdot \mathrm{RMSD}^2\left(\mathbf{M}, \vec{t}\right) = \sum_{(\vec{u},\vec{v})\in\mathbf{M}} \left\langle \vec{t},\vec{t}\right\rangle + 2\left\langle \vec{u},\vec{t}\right\rangle - 2\left\langle \vec{v},\vec{t}\right\rangle + \left\langle \vec{u},\vec{u}\right\rangle + \left\langle \vec{v},\vec{v}\right\rangle - 2\left\langle \vec{u},\vec{v}\right\rangle \quad (6.5)$$

It can be seen that, for any matching, the squared RMSD is a multivariate quadratic function of $\vec{t}$. The matching-dependent term is independent of $\vec{t}$. As such, a matching with the lowest RMSD at a given translation also has the lowest RMSD at *every* translation. □

Lemma 2. *Consider a matching of two (non-periodic) point sets* $\mathbf{U}$ *and* $\mathbf{V}$ *and a translation* $\vec{t}$ *of* $\mathbf{U}$*. The optimal translation is the one which brings the barycentre of* $\mathbf{U}$ *onto the barycentre of* $\mathbf{V}$*.*

Proof. This is a well-known result. Setting the gradient of equation (6.5) to zero and rearranging gives:

$$\nabla\mathrm{RMSD}^2\left(\mathbf{M}, \vec{t}\right) = 0 \quad \Rightarrow \quad \vec{t}^{\,*} = \frac{1}{n}\sum_{(\vec{u},\vec{v})\in\mathbf{M}} \vec{v} - \vec{u} \quad (6.6)$$

Note that the optimal translation is an invariant of the point sets (it is independent of the matching). □

By Lemmas 1 and 2 there is only one optimal matching between two non-periodic point sets. We now consider crystal lattices.

Theorem 1. *Consider two periodic point sets* $\mathbf{U}$ *and* $\mathbf{V}$*, both in* $\mathbb{R}^d$*, and a translation* $\vec{t}$ *of* $\mathbf{U}$*. The number of minimizing matchings is at most* n^d*.*

Proof. By combining equations (6.2) and (6.6) it can be shown that an expression for the optimal translation of any matching must be of the form:

$$\vec{t}^{\,*}_{\mathrm{lattice}} = \frac{1}{n}\left[\mathbf{C}^T\vec{h} + \sum_{(\vec{u},\vec{v})\in\mathbf{M}} \vec{v} - \vec{u}\right] \quad (6.7)$$

where $\vec{h} \in \mathbb{Z}^d$. Due to periodicity we need only consider $\vec{h} \in \{0, 1, 2, \ldots, n-1\}^d$. Therefore, there are at most n^d optimal translations. By Lemma 1 there cannot be more minimizing matchings than optimal translations. □

6.4 Illustration of Principles

Theorem 1 is more easily understood with a visual example. Figure 6.3 shows two periodic 1D lattices, and the associated squared RMSD curve at every translation t. The figure illustrates the result of Theorem 1, namely that there are no more than n minimizing matchings. In the 1D case, each curve corresponds to a matching with no edge crossings. Figure 6.4 shows a 2D example, where the translation now has two components.

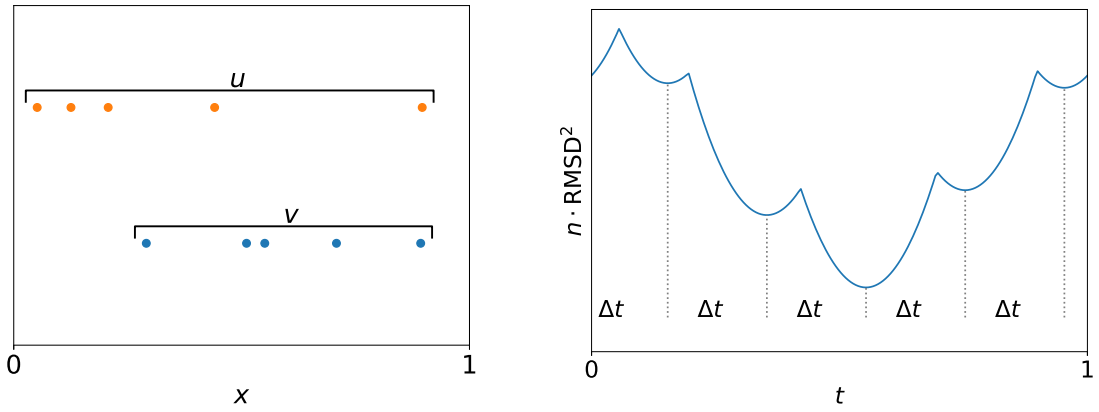


Figure 6.3. Illustration of the number of matchings in a 1D periodic lattice. **Left:** two point sets, **U** and **V**, constitute the atomic bases of the two crystals. The point sets are drawn on top of each other for illustration purposes only; they are 1D point sets and have an x-component only. The lattice parameter is simply 1. **Right:** The optimal RMSD between **U** and **V** as a function of the translation, t. The curve of each minimizing matching, of which there are exactly n, is a parabola. The minima of the parabola are separated by $\Delta t = \frac{1}{n}$.

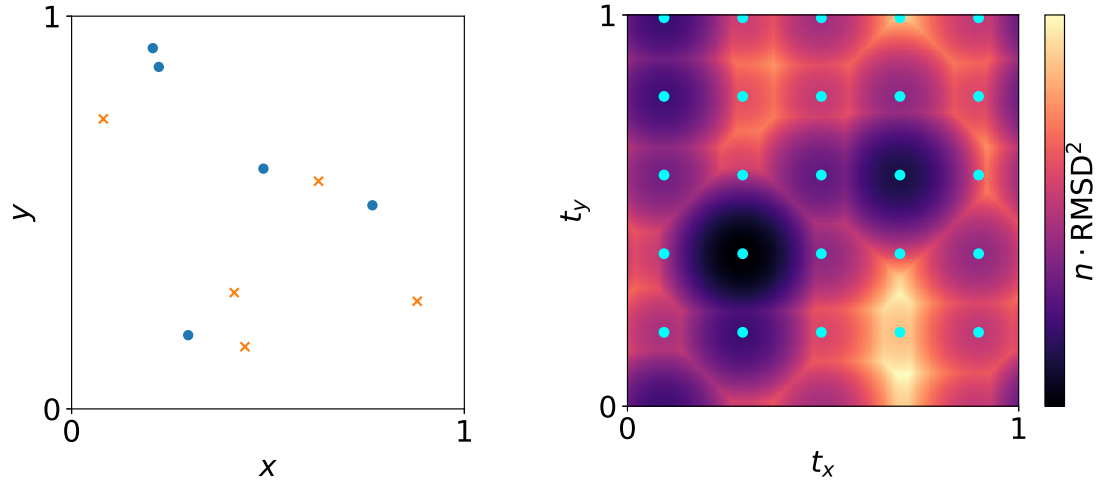


Figure 6.4. Illustration of the number of matchings in a 2D periodic lattice. **Left:** two point sets, **U** and **V**, constitute the atomic bases of the two crystals. In this example, the lattice vectors are $\vec{a} = \{0, 1\}$ and $\vec{b} = \{1, 0\}$, though the principles apply equally to a non-orthogonal set of lattice vectors and/or lattice vectors of different lengths. **Right:** The optimal RMSD between **U** and **V** as a function of the translation, $\vec{t}$. The squared-RMSD of each minimizing matching is a multivariate quadratic function of $\vec{t}$, of which there are exactly n^2. The minima (marked with cyan circles) are separated by $\Delta t = \frac{1}{n}$ in each direction.

6.5 An Algorithm to Compute all Minimizing Matchings

We have shown how to calculate the RMSD between two lattices given that they have the same number of atoms and the same unit cells. Practically, though, no two materials have the same unit cells. We need to define a mapping such that two different unit cells are brought into the same frame of reference. We can achieve this quite simply using the affine registration method described in section 3.2.

Given two unit cells $\mathbf{C}_1$ and $\mathbf{C}_2$, both in $\mathbb{R}^3$, with lattice vectors $\vec{a}_1, \vec{b}_1, \vec{c}_1$ and $\vec{a}_2, \vec{b}_2, \vec{c}_2$, we can find a linear map $\mathbf{A}$ which maps one unit cell onto the other:

$$\mathbf{A} \begin{bmatrix} \vec{a_1} \\ \vec{b_1} \\ \vec{c_1} \end{bmatrix} = \begin{bmatrix} \vec{a_2} \\ \vec{b_2} \\ \vec{c_2} \end{bmatrix} \tag{6.8}$$

In this form, the linear map, and consequently the RMSD, depends upon which unit cell is to be mapped onto the other. A better choice is to map each unit cell such that they meet 'half-way', in an intermediate unit cell. The half-way mapping is given by $\frac{1}{2}(\mathbf{I} + \mathbf{A})$, and the intermediate unit cell, $\mathbf{F}$, is then:

$$\mathbf{F} = \frac{1}{2}(\mathbf{I} + \mathbf{A}) \begin{bmatrix} \vec{a_1} \\ \vec{b_1} \\ \vec{c_1} \end{bmatrix} = \frac{1}{2}\left(\mathbf{I} + \mathbf{A}^{-1}\right) \begin{bmatrix} \vec{a_2} \\ \vec{b_2} \\ \vec{c_2} \end{bmatrix} \tag{6.9}$$

Since the mapping is linear, the intermediate cell can be calculated more simply as:

$$\mathbf{F} = \frac{1}{2}(\mathbf{C}_1 + \mathbf{C}_2) \tag{6.10}$$

A requirement for the calculations in Equations (6.9) and (6.10) is that the unit cells are lower-triangular form, that is:

$$\mathbf{C} = \begin{bmatrix} \vec{a}_x & 0 & 0 \\ \vec{b}_x & \vec{b}_y & 0 \\ \vec{c}_x & \vec{c}_y & \vec{c}_z \end{bmatrix} \tag{6.11}$$

If this is not the case, a QR decomposition will produce a correctly formed unit cell.

The mapping into the intermediate cell is illustrated in Figure 6.5, for a single mapping. Depending on the space group, there are up to 48 different permissible mappings from one cell onto the other, and the same number of corresponding intermediate cell mappings. In order to find the minimum RMSD, all permissible mappings must be investigated.

We can now state the full process for calculating the RMSD between two lattices, which is shown in 6.1. By Theorem 1, the number of minimizing matchings is n^d. The most expensive operation in the inner loop of 6.1 is the solution of the bipartite graph matching problem. The Hungarian algorithm can solve this in $O(n^3)$, giving a total running time of $O\left(n^{d+3}\right)$. In the expected use case of $\mathbb{R}^3$, a running time of $O(n^6)$ is

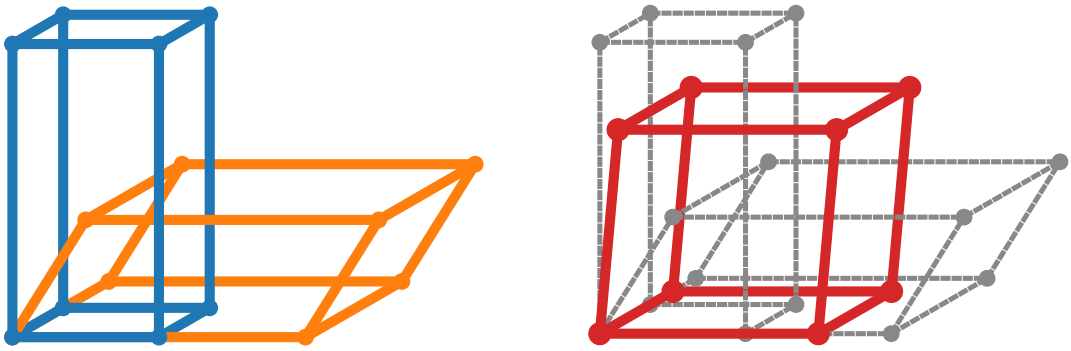


Figure 6.5. Left: Two unit cells which must be mapped into the same frame of reference in order to calculate the RMSD of their atomic content. The blue cell can be mapped onto the orange cell, or vice versa. **Right:** A better choice is to map both cells such that they meet half-way, shown by the red cell. This obviates the need to choose a mapping direction.

clearly expensive, and exhibits poor scaling with increasing number of atoms. It is likely that the dynamic Hungarian algorithm [64] can be used instead of the classic Hungarian algorithm to reduce the running time to $O\left(n^{d+2}\right)$, but this remains to be investigated.

Despite the relatively poor scaling of the algorithm presented here ($O(n^5)$ is rarely considered fast), it represents an enormous improvement upon the existing factorial-scaling methods. For a system of 40 atoms, an unoptimized version of the method presented here requires of the order 1 minute to calculate the RMSD. In a factorial scaling algorithm, $40! \approx 10^{47}$ comparisons are required; the lifetime of the universe is insufficient to perform such a number of calculations on the current fastest supercomputer.

Furthermore, we note that, in the mathematical community, the first description of a polynomial time algorithm for any problem has rarely achieved the best possible running time, but has been subsequently improved upon by other researchers. Our result bounds the running time from below to $O\left(n^d\right)$ (the number of minimizing matchings), a running time which we cannot rule out. Achieving this bound would be a significant improvement, and would make the calculation of the RMSD significantly more practical, regardless of system size.

Algorithm 6.1 Pseudocode for determining the optimal RMSD between two lattices. The algorithm shown here is for a monoatomic basis (one ony element), but can be easily adapted for lattices with multiple elements.

procedure DETERMINESTRUCTURE($\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{U}_{\text{frac}}$, $\mathbf{V}_{\text{frac}}$)
▷ $\mathbf{C}_1$: unit cell of the first lattice
▷ $\mathbf{C}_2$: unit cell of the second lattice
▷ $\mathbf{U}_{\text{frac}}$: fractional coordinates of first lattice
▷ $\mathbf{V}_{\text{frac}}$: fractional coordinates of first lattice
for each permissible mapping of the space group **do**
$\mathbf{F} := (\mathbf{C}_1 + \mathbf{C}_2)$ ▷ Calculate intermediate cell
$\mathbf{U} := \mathbf{F}\mathbf{U}_{\text{frac}}$ ▷ Map fractional coordinates into intermediate cell
$\mathbf{U} := \mathbf{U} - \overline{\mathbf{U}}$ ▷ Subtract barycentre
$\mathbf{U} := \mathbf{U} \mod \mathbf{F}$ ▷ Map atoms into cell using periodic boundary conditions
(Do the same for $\mathbf{V}$)
$\text{RMSD}^* = \infty$
for $k \in \{0, 1, \ldots, n-1\}$ **do**
for $j \in \{0, 1, \ldots, n-1\}$ **do**
for $i \in \{0, 1, \ldots, n-1\}$ **do**
$\mathbf{D}_{\vec{u},\vec{v}} = \arg\min_{\vec{h}} \left|\left|\vec{u} - \left(\vec{v} + \mathbf{F}^T\vec{h}\right)\right|\right|$ ▷ Calculate distance matrix
Find matching $\mathbf{M}$ by solving bipartite graph matching problem
Calculate RMSD (c.f. Equation(6.2))
$\text{RMSD}^* = \min(\text{RMSD}^*, \text{RMSD})$ ▷ Save best RMSD value
$\mathbf{U} := \left(\mathbf{U} + \frac{1}{n} i \cdot \vec{a}\right) \mod \mathbf{F}$ ▷ Move onto next basin
end for
$\mathbf{U} := \left(\mathbf{U} + \frac{1}{n} j \cdot \vec{b}\right) \mod \mathbf{F}$ ▷ Move onto next basin
end for
$\mathbf{U} := \left(\mathbf{U} + \frac{1}{n} k \cdot \vec{c}\right) \mod \mathbf{F}$ ▷ Move onto next basin
end for
end for
return RMSD^*
end procedure

6.6 Quantifying the Geometric Difference Between Unit Cells

We have described a method for calculating the RMSD between two lattices when mapped *into a common frame of reference*. An obvious shortcoming is the case where two cells have the same fractional coordinates but different unit cells, as shown in Figure 6.6. The cells are clearly different, and the change in cell geometry, which is equivalent to a strain being applied, will in most cases result in a change in energy. We can quantify the change in cell geometry using the methods described in section 3.

Let $\mathbf{A}$ and $\mathbf{B}$ be the two cells we wish to compare, and let $\mathbf{M}$ be the linear map:

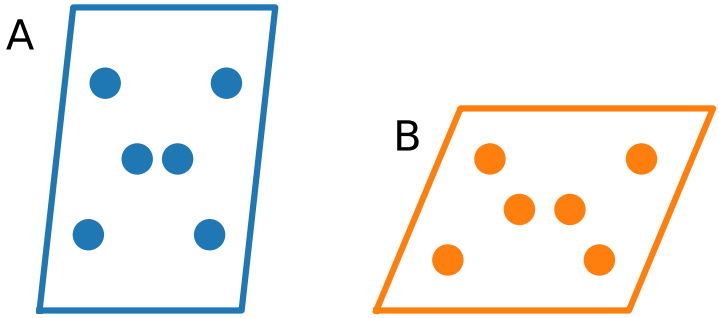


Figure 6.6. Two unit cells with different geometries but whose atomic bases have the same fractional coordinates. The RMSD between the two materials is zero, yet clearly they are different. The difference in cell geometry must also be quantified.

$\mathbf{MA} = \mathbf{B}$. Next, take a polar decomposition $\mathbf{QP} = \mathbf{M}$, such that $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{P}$ is a positive-definite symmetric matrix (a strain matrix). Then, the change in cell geometry can be 'boiled down' to a scalar by:

$$\Delta_{\text{cell}} = ||\mathbf{P} - \mathbf{I}||_{\mathbf{F}} \tag{6.12}$$

where $\mathbf{I}$ is the identity matrix, and $||\ldots||_{\mathbf{F}}$ denotes the Frobenius norm. The above equation measures the difference between the strain matrix and the identity matrix. Since the identify matrix corresponds to zero strain, Δ_{cell} is zero when $\mathbf{P} = \mathbf{I}$. The polar decomposition of $\mathbf{M}$ ensures that cell rotations are not included in the measure.

6.7 Practical Application

Here, we briefly demonstrate the use of the method for identifying prototype clusters. We have extracted a set of 151 structures from the ICSD [49], all of which have space group 14 and the abstract formula A_8B_4. Since they have the same stoichiometry and space group, they potentially have the same crystallographic prototype. Figure 6.7 shows the results of the analysis. The RMSD has been calculated between each pair of structures. The resulting distance matrix has been optimally permuted using the rearrangement clustering method [65], such that the clusters are revealed. It can be seen that there are three large prototype clusters, and a range of smaller clusters. The single dark pixels which lie on the matrix diagonal are unique structures. Clustering in this way reveals the relationship between the structures. We emphasize that this has been achieved without any parameters needing to be specified.

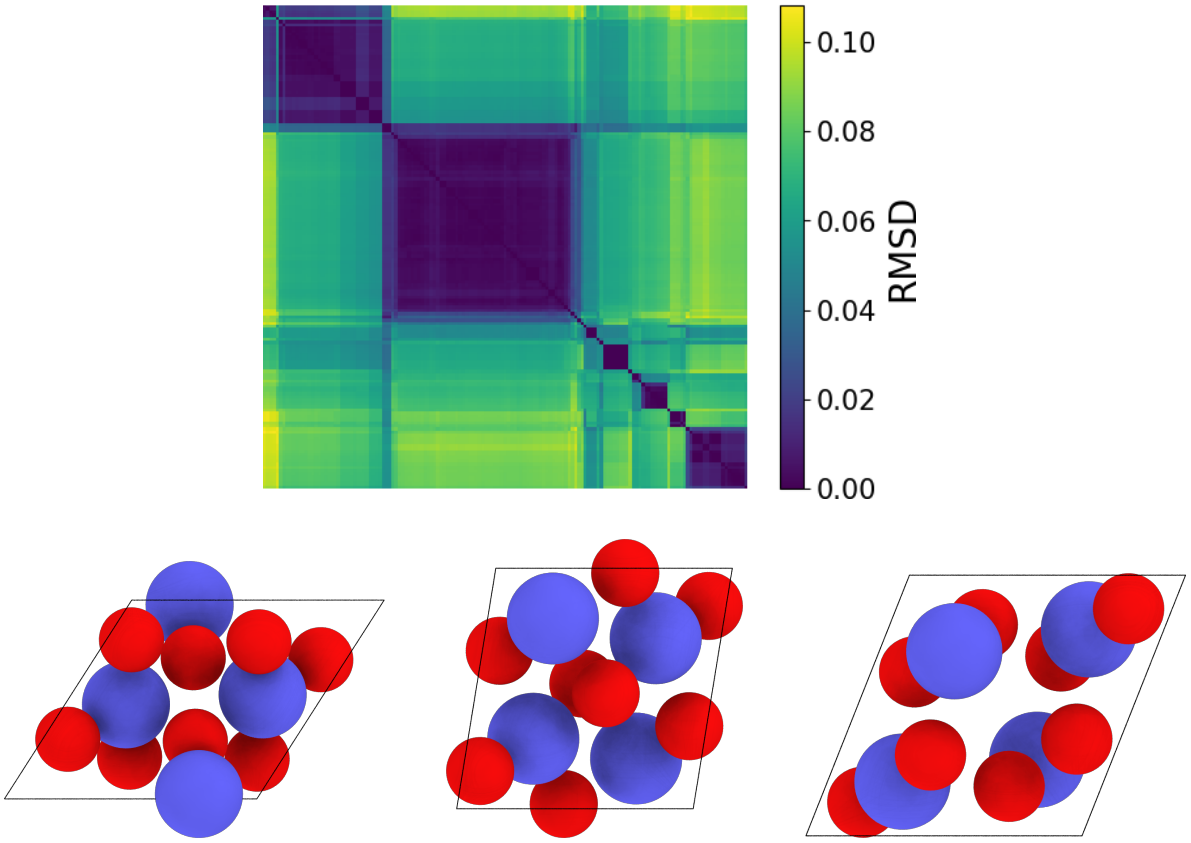


Figure 6.7. Top: Inter-structure distance matrix of all 151 structures in the ICSD with space group 14 and abstract chemical formula A_8B_4. The structures have been ordered by similarity, revealing thre large prototype clusters. **Bottom:** The structures of the largest prototype clusters.

7 A Metric for 2D Monolayers

A relatively new problem for materials scientists is the comparison and classification of 2D materials. Novoselov and Geim's 2004 production and characterization of monolayer graphene [66] is the genesis of the current interest in 2D materials. Due to their relative novelty, 2D materials have not received the same attention as bulk materials with respect to structural comparison methods. For the same reason, there is a pressing need for a metric in order to identify duplicate structures, assign prototypes, and follow the structural evolution in the simulation of 2D monolayers. Here, we describe an extension to 2D materials of the RMSD calculation method described in section 6.

In the RMSD calculation for *bulk* materials, the two unit cells were mapped into the same frame of reference (an intermediate unit cell). A simple method was described for finding the optimal translation, by translating the atomic basis of one lattice along all three periodic directions. Materials in a 2D monolayer pose an extra challenge, since they are not periodic in the out-of-plane direction. As we shall show, the artificial imposition of periodicity is not meaningful either.

7.1 Mapping into a Common Frame of Reference

Figure 7.1 shows a MoS_2 monolayer. There are two in-plane lattice vectors, $\vec{a}$ and $\vec{b}$, which specify the two periodic directions of the material. If the material is truly two-dimensional, such as e.g. graphene, we could simply discard $\vec{c}$, the out-of-plane vector, and reuse the bulk RMSD calculation for $\mathbb{R}^2$. Most 2D materials, however, have atomic positions with an out-of-plane component, such as the sulphur atoms shown in blue in Figure 7.1.

Many databases and structure files nonetheless specify $\vec{c}$, the out-of-plane vector. This serves two purposes. Firstly, DFT energy calculations with a plane-wave basis require periodicity along all lattice vector directions, so a periodic cell with a large $\vec{c}$ is used to approximate a monolayer. Secondly, a cell with all three lattice vectors permits the calculation of fractional coordinates. The fractional coordinates, though, cannot be used when comparing two different 2D materials; if the length of $\vec{c}$ is doubled and the

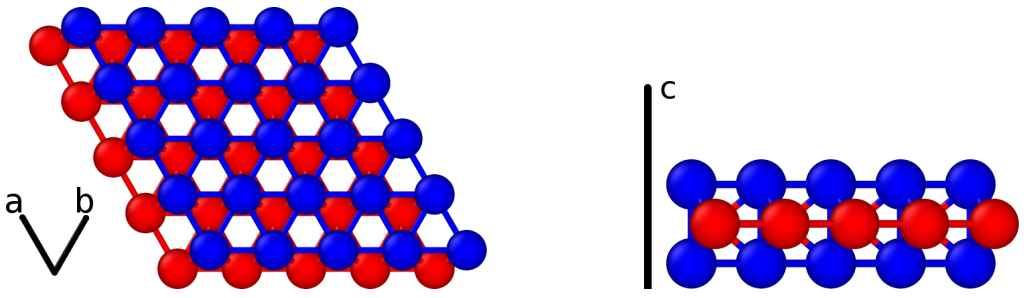


Figure 7.1. Structure of a MoS_2 monolayer. **Left:** Top view. The two in-plane lattice vectors are well defined. **Right:** Side view. The out-of-plane 'lattice vector' has no meaning in a monolayer context, since there is no periodicity in this direction.

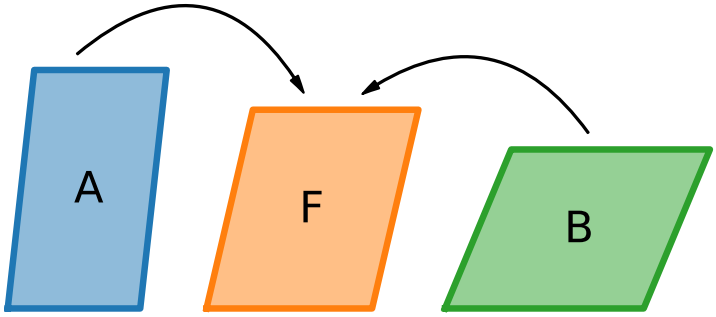


Figure 7.2. Mapping of the atomic content from two 2D monolayers into the same frame of reference. A linear map is used to map cells A and B into an intermediate cell, F. The intermediate cell is given by $\mathrm{F} = (\mathrm{A} + \mathrm{B})/2$.

corresponding fractional coordinates halved, the resulting structure is identical. Since the length of $\vec{c}$ is arbitrary, mapping two materials into the same frame of reference as in the bulk RMSD calculation would introduce an arbitrary 'squashing factor' in the out-of-plane direction.

Before we can calculate the RMSD, we first need to map both materials into the same frame of reference. This is illustrated in Figure 7.2. Only the in-plane vectors are used to determine the intermediate cell, i.e. the mapping is a 2D transformation. We extend the linear map to $\mathbb{R}^3$ as follows. Let κ_A, κ_B and κ_F be the areas of the 2D cells. Then $\kappa_A = \left|\left|\vec{a}_A \times \vec{b}_A\right|\right|$, where $\vec{a}_A$ and $\vec{b}_A$ are the in-plane lattice vectors of cell A and $\times$ denotes the cross product (this is just the area formula for a parallelogram). To extend the linear map to $\mathbb{R}^3$ we need to scale the atomic coordinates in the out-of-plane direction. The only natural scaling factors are $\sqrt{\kappa_F/\kappa_A}$ and $\sqrt{\kappa_F/\kappa_B}$ for the maps which bring respectively A and B onto F. With the atomic coordinates mapped into a common frame of reference, we need to determine the translation which minimizes the RMSD, and, optionally, optimize the scaling in the out-of-plane direction.

7.2 Calculation of the RMSD

In the bulk RMSD calculation, a translation was found in $\mathbb{R}^3$ which minimizes the RMSD. In a 2D monolayer, we calculate an optimal translation in $\mathbb{R}^2$, and, optionally, an optimal scaling in the out-of-plane direction. Given two materials with a significant out-of-plane component, it may be desirable to determine whether they differ in the out-of-plane direction only, c.f. Figure 7.3. If this is the case, we wish to scale the materials such that this difference is removed. The scaling here is not the same as the scaling used to map the materials into the common frame of reference (described above); in the latter case, each material is scaled independently. Here, we apply a common, symmetric scaling factor. We can adapt Equation (6.2) to define the RMSD between two monolayer materials, using a symmetric scaling in the out-of-plane direction:

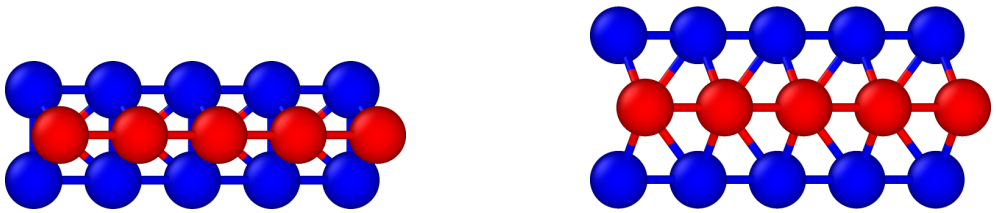


Figure 7.3. Two 2D monolayer materials whose geometries differ in the out-of-plane scale only. For applications such as the assignment of crystallographic prototypes, it is desirable to calculate the RMSD 'modulo scaling'.

$$\mathrm{RMSD}_{\mathrm{monolayer}}(\mathbf{M}, \vec{t}, s) = \sqrt{\frac{1}{n} \sum_{(\vec{u},\vec{v})\in\mathbf{M}} \arg\min_{\vec{h}} \left|\left|\mathbf{S}\left(\vec{u}+\vec{t}\right) - \mathbf{S}^{-1}\left(\vec{v}+\mathbf{C}^T\vec{h}\right)\right|\right|^2} \tag{7.1}$$

where $\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix}$ is a matrix which scales the point sets $\mathbf{U}$ and $\mathbf{V}$ in the out-of-plane direction, and s is the symmetric scaling factor. Then, the most natural choice of scaling factor is simply the one which minimizes the RMSD:

$$d\left(\mathbf{U}, \mathbf{V}\right) = \arg\min_{\mathbf{M},\vec{t},s} \mathrm{RMSD}_{\mathrm{monolayer}}\left(\mathbf{M}, \vec{t}, s\right) \tag{7.2}$$

Let $\vec{u}_z$ and $\vec{v}_z$ denote the out-of-plane ordinates of the atoms in each matched pair. Then, the contribution of the out-of-plane direction to the RMSD is given by:

$$\sum_{\vec{u},\vec{v}\in\mathbf{M}} \left(s\vec{u}_z - \vec{v}_z/s\right)^2 = \sum_{\vec{u},\vec{v}\in\mathbf{M}} s^2\vec{u}_z^2 - 2\vec{u}_z\vec{v}_z + \vec{v}_z^2/s^2 \tag{7.3}$$

By differentiation and rearrangement of Equation (7.3) it can be shown that the optimal scaling factor is given by:

$$s^* = \sum_{\vec{v}\in\mathbf{V}} \vec{v}_z^2 \Big/ \sum_{\vec{u}\in\mathbf{U}} \vec{u}_z^2 \tag{7.4}$$

Quite remarkably, the optimal scaling factor is independent of the matching. As such, we can calculate the optimal scaling factor before finding the optimal in-plane translation, and in linear time. This scaling is ideal for finding the RMSD in monolayers with a significant thickness, but should not be used when at least one monolayer is a perfect 2D material. For example, in a comparison involving graphene, whose out-of-plane components are zero, Equation (7.4) is either singular (since the denominator is zero), or calculates an optimal scaling factor of zero. In the latter case, the second structure is 'pancaked', by which we mean that the scaled out-of-plane ordinates are zero. As such, the choice of whether to calculate an optimal scaling depends upon the application and the materials involved.

Combining all the steps, we can describe the algorithm to calculate the RMSD in 2D monolayers:

1 Calculate a common frame of reference (the intermediate cell) in $\mathbb{R}^2$.
2 Extend the linear map to $\mathbb{R}^3$ using the scaling factors $\sqrt{\kappa_F/\kappa_A}$ and $\sqrt{\kappa_F/\kappa_B}$.
3 If desired, optimize the out-of-plane scaling using Equation (7.4).
4 Find the optimal translation in $\mathbb{R}^2$ using the bulk RMSD method described in section 6.

7.3 Quantifying the Geometric Evolution in Structural Relaxation of Monolayers

We now provide an example use-case of the method. Ab-initio calculations are commonly used to identify stable 2D monolayers. A large supercell is created, and the energy of the system is (locally) minimized by optimizing the atomic positions, usually using a variant of gradient descent. If the atomic positions of the supercell are largely identical to the repeated unit cell, the monolayer is likely a stable structure. Conversely, if the atomic positions in the supercell differ significantly from those of the repeated unit cell, the single unit cell is not a stable structure.

One would expect that a large change in atomic positions would imply a large change in the total energy. Whilst this is often the case, some monolayers have a very 'shallow' energy potential, whereby the atoms can move far from their original positions without a large change in energy. In this case, the change in total energy is not a good indicator of structural evolution.

Given a series of relaxation steps, we can follow the structural evolution by calculating the RMSD between the first step and each subsequent steps. We have done so for three different monolayer materials, $AlCl_2$, $RhCl_2$ and $PtBr_2$† The structural evolution is shown in Figures 7.4 and 7.5.

The magnitude of the changes in the structure is clearly reflected by the RMSD. Another notable observation is that the RMSD does not increase monotonically. In some cases the RMSD decreases consistently over multiple relaxation steps, whereas any good structural relaxation method should not permit the total energy to consistently increase over multiple steps. This demonstrates the lack of a one-to-one relationship between geometry and total energy. We intuitively expect this, since the energy has a complex dependency on the atomic positions. Nonetheless, when the structures are fully relaxed and the energy converges, the RMSD also stabilizes.

† Monolayer structures kindly provided by M. Pandey.

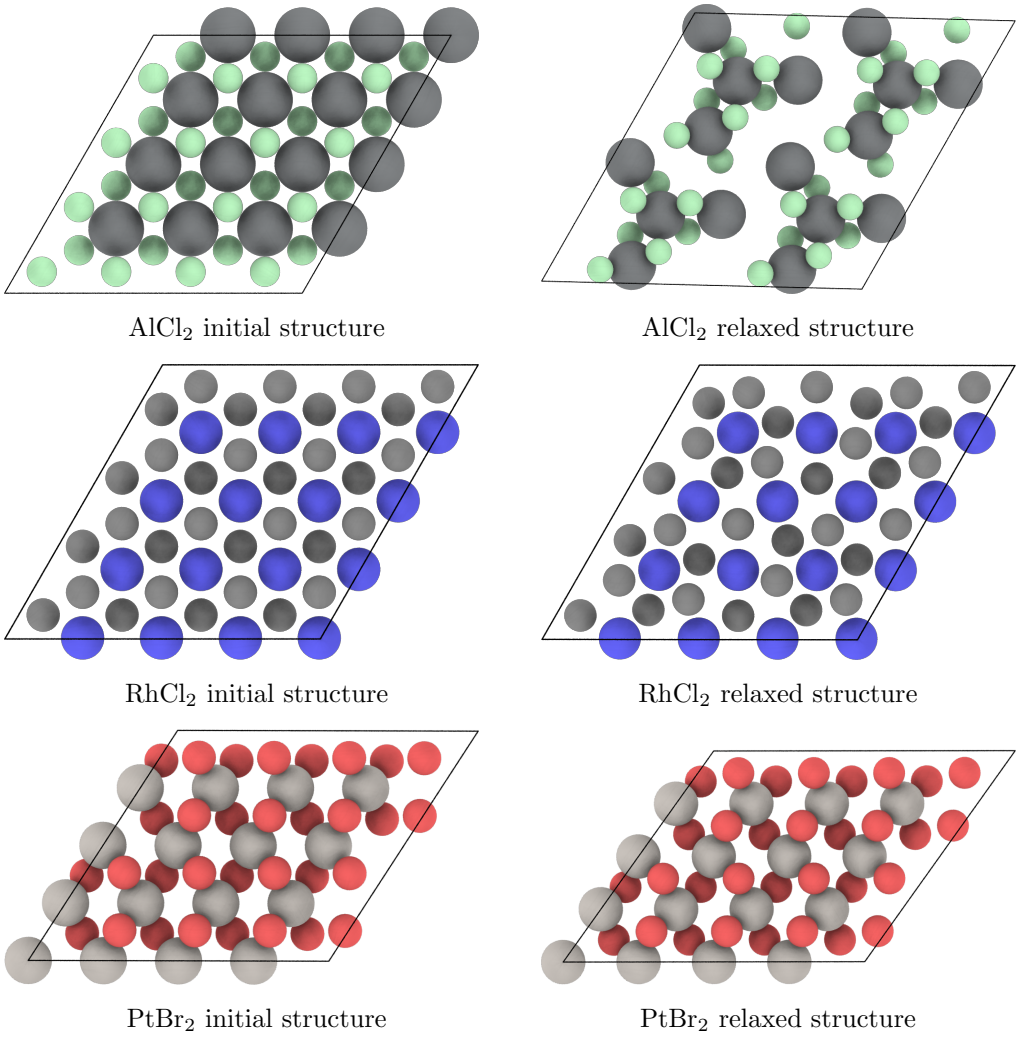


Figure 7.4. Structural relaxation of three 2D monolayer materials, exhibiting varying degrees of geometric evolution.

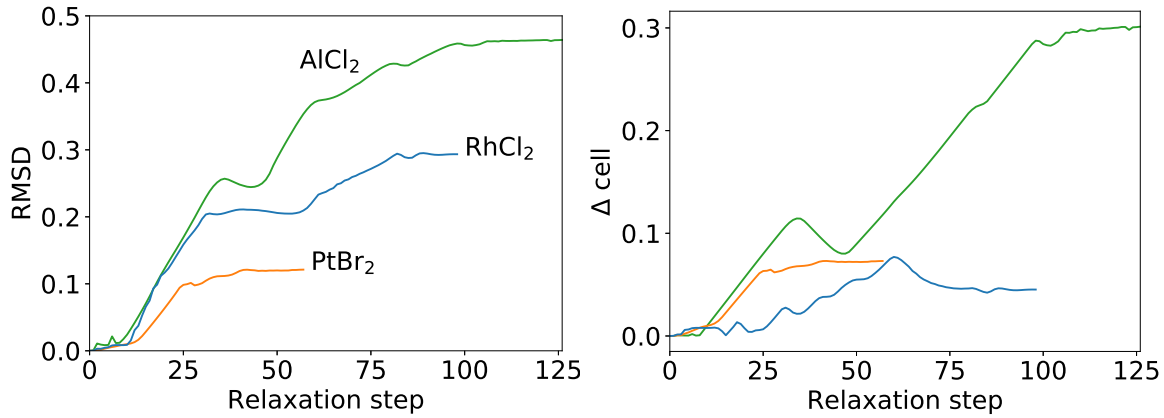


Figure 7.5. Left: RMSD of each monolayer at every relaxation step. The RMSD is not a bijective function of the total energy, as shown by the intervals where the RMSD consistently decreases. **Right:** Evolution in unit cell geometry, as quantified by Equation (6.12).

8 Identification of Lattice Phases

In the previous sections we have used graph matching to calculate centrosymmetry parameters and to measure the RMSD in lattices and 2D monolayers. We will now apply these techniques to the problem of determining local structures in molecular dynamics simulations.

A molecular dynamics simulation consists of a simulation cell, atomic numbers, atomic coordinates, and kinetic and potential energies for each atom. Early attempts at structural classification used only the energy of the atoms in a quenched state, since crystalline atoms have a lower energy than defect atoms. This method, however, does not permit to distinguish between different types of crystalline structures and/or different types of defects. Later methods, including CSP [37], Common Neighbour Analysis [38] (CNA), Steinhardt bond order parameters [67], and Voronoi-based identification [68], use only the atomic coordinates to determine structure.

Given a bond length parameter, CNA defines a neighbour relationship between all atoms whose distance is less than the bond length. Thereafter, a signature is computed based on the number of neighbours each atom has in common with other atoms (hence the name). This signature is compared against a set of precomputed signatures for each structure type. Interestingly, whilst not formulated as such, this is in fact a graph isomorphism problem; the atoms are vertices, the bonds are edges, and the comparison of the signature against a predetermined set is a home-made graph isomorphism test.

The Voronoi-based method of Lazar et al. [68] constructs a planar graph of the Voronoi cell of each atom, and compares it against a predetermined set of graphs for each structure; this is also a graph isomorphism problem and explicitly recognized as such. Once the graph of the Voronoi cell is constructed, the problem is topological only, since the atomic coordinates are no longer used. In some cases, this limits the ability to distinguish geometrically different structures whose Voronoi graphs are identical.

We have developed a method, Polyhedral Template Matching (PTM, presented in Paper 1), which combines geometric and topological information to determine local structure. For each structural type which we wish to identify, we can construct a template using the nearest neighbours of a central atom, as shown in Figure 8.1. Whilst

Figure 8.1. Templates for the five different crystalline structures which PTM identifies. The template vertices consist of the the nearest neighbours of an atom in a perfect lattice. The BCC structure requires the atoms of the first two neighbour shells for robust matching. Figure reproduced from Paper 1.

there are many details in the algorithm, the problem can be expressed very succinctly: given a central atom and its nearest neighbours, which template (if any) fits best? The measure of how well a template fits is simply the RMSD. Once a match has been found, many properties such as the local orientation, strain, and alloy ordering can be determined, but for these details we refer to the paper. Here, we focus on the graph matching approach and compare it to the approaches presented in previous sections. We will first review the different types of geometric graph problems presented so far.

8.1 Overview

In the preceding sections we have described a range of geometric matching problems, with a range of different restrictions. For example, in the standard bipartite matching problem in geometric graphs, as presented in section 2, the translation and rotation are fixed, and only the matching is a variable. In the lattice matching problem, presented in the last two sections, the rotation is implicitly fixed by the mapping into an intermediate cell, and the variables are the translation and the matching. Due to certain variables being fixed, each of these problems permits a polynomial-time solution. Table 8.1 shows the parameters of each matching problem, and the associated algorithmic complexities. For the present problem, none of the variables are fixed, a problem for which no polynomial-time algorithm is known.

In section 6 we showed that measurement of the RMSD between lattices can be formulated as a bipartite matching problem, where the edge weights vary as a function of translation. For that application there are two naive solutions for determining the optimal matching. The first is to test every possible matching, of which there are factorially many. The second is to test every possible translation, of which there are infinitely many. However, by deriving a constraint on locally optimal translations, the number of relevant tests is reduced to a polynomial number.

Method	Fixed matching	Fixed translation	Fixed rotation	Alg. complexity
Optimal rotation determination	✓	✗	✗	$O(n)$
Geometric bipartite graph matching	✗	✗	✓	$O(n^3)$
Lattice matching	✗	✗	✓	$O\left(n^{d+3}\right)$
Template matching (present work)	✗	✗	✗	$O(n!)$

Table 8.1. Comparison of geometric matching problems presented in this thesis. The fixed parameters are either the matching, translation, or rotation. With the exception of the lattice matching problem (in which the translation is periodic) the translation can be determined a priori (c.f. section 3). No polynomial-time algorithm is known for geometric matching with neither a fixed matching nor a fixed rotation.

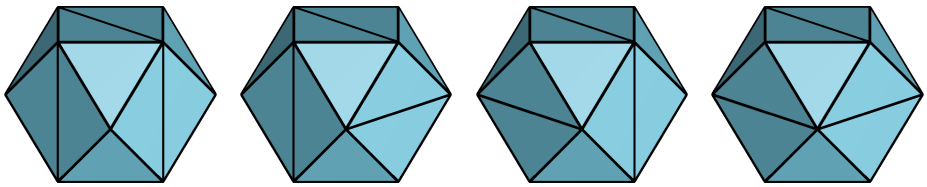


Figure 8.2. The convex hull of the FCC template contains six square facets, each of which has two different triangulations. In total, the convex hull has 2^6 different triangulations. Figure reproduced from Paper 1.

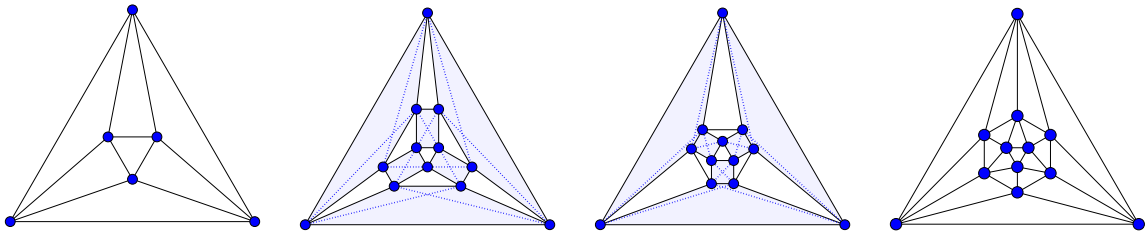


Figure 8.3. The planar graphs of the template convex hulls. Dotted lines represent multiple triangulations of square or rhombus facets. Figure reproduced from Paper 1.

In this application, we must match a template against a set of simulated atomic coordinates. This is also a weighted bipartite matching problem, where the weights now vary as a function of the template *orientation*. The naive solutions are similarly, to test every matching, or to test every possible template orientation. Unlike the crystal lattice matching problem, for which the translations could be constrained, in the orientational case no such constraint exists in the literature, nor could we derive one. Instead, we use graph theoretical considerations to reduce the number of relevant matchings from factorially many to a *small* number. This solution is not as elegant, since we derive no general upper bound on the number of relevant matchings, but in practice it is very efficient.

8.2 Approach

Consider the polyhedral templates in Figure 8.1; all of them are convex. This permits a natural representation of both the templates and the simulated nearest neighbours in graph form: compute the (triangulated) convex hull, producing a graph whose vertices are nearest neighbour atoms and whose edges are given by the convex hull. For most templates, multiple triangulations of the convex hull are possible, with infinitesmal perturbations being sufficient to change the triangulation (c.f. Figure 8.2). As such, we precompute all possible triangulations of each template. By Steinitz' theorem [69], the graph of a convex hull is planar. Exploiting this fact, we can represent all possible template graphs as drawings in the plane, as shown in Figure 8.3.

When determining the structure of a simulated atom, we obtain a graph from the convex hull of its nearest neighbours, as described above. Given a matching, the optimal

orientation (and RMSD) is easily determined, as discussed in section 3.3. Thus, for every template whose graph representation is isomorphic to the simulated atom's graph, the set of matchings to investigate is limited to the graph *automorphisms*. We repeat this process for every number of nearest neighbours in the set of templates (6 for SC, 12 for FCC, HCP and ICO, and 14 for BCC) and select the matching with the lowest RMSD.

This approach results in a significant (huge) reduction in the number of matchings to be investigated, from factorially many to at most 10. The graph representation is only used to rule out some structures (those which are not isomorphic to a template graph) and quickly suggest possible matchings, whereafter the template with the lowest RMSD determines the best matching.

If the structure is not isomorphic to any template graph, it is assigned an RMSD of infinity; this means that the method does not constitute a true metric. Given two structures with same number of vertices in a finite sized simulation cell, an infinite RMSD is clearly nonsensical; a disordered atom should still have an optimal match, albeit with a very high RMSD. An RMSD threshold could then be used to discriminate between defect atoms and actual crystalline atoms † Nonetheless, this does not represent a barrier to the method's use, as the distinction between high-RMSD atoms and disordered atoms is not a practical concern.

Usage examples are demonstrated in Paper 1. Other successful applications of the method in the literature include modelling new deformation mechanisms in nanocrystals [71], dislocation and misorientation dependence of grain growth [72], and abrasion of nanocrystalline ferrite [73].

8.3 Structural Analysis of 2D Materials with Polygonal Template Matching

The PTM method we have presented is developed for MD simulations in three dimensions, but the concepts generalize very well to two-dimensional materials. Here, we demonstrate the analysis of simulated graphene sheets. Rather than calculate geometric properties (structural type, RMSD, orientation, strain) on a per-atom basis, we calculate them for each 'hole' between carbon rings, as this is visually more informative for graphene sheets.

Perfectly crystalline graphene has remarkable mechanical and electronic properties. Any industrial production method, though, invariably results in multiple crystallites, i.e. polycrystalline graphene. The interface between two crystallites, known as a grain boundary, will in most cases contain non-hexagonal structures. Other types of defects include point defects and dislocations (for a good overview we refer to the review paper by Banhart et al. [75]). The types and distributions of defects in pure graphene determines the degradation in mechanical properties. On the other hand, the presence

† Whilst I consider this to be a flaw of the method, I have found examples of people using it as a feature. The fact that disordered atoms are often anisomorphic to any template graph is equivalent to selecting a high RMSD threshold. As such, it has even been described as a parameter-free method [70]. Nonetheless, this is not how I intended the method to be used.

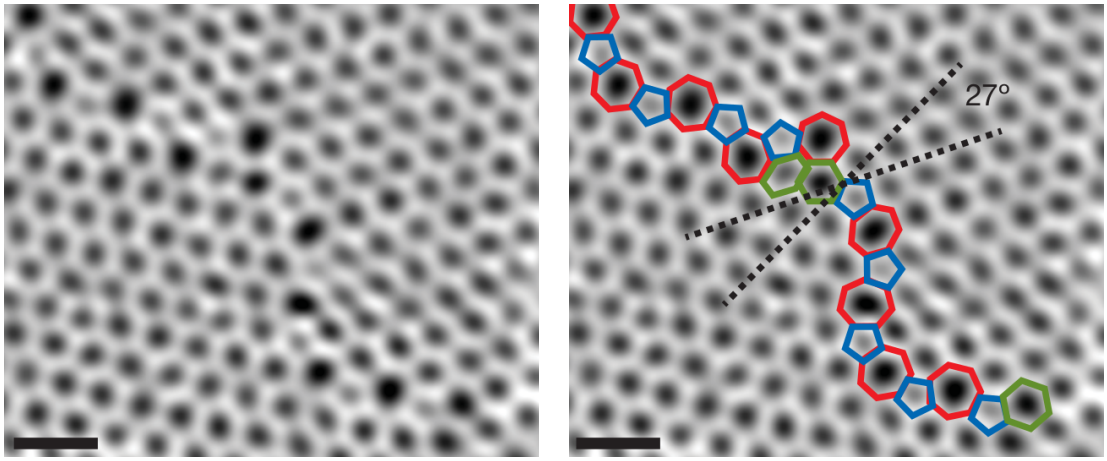


Figure 8.4. Left: Grain boundary between two graphene crystallites, observed in a TEM. **Right:** Highlighted defect structures. The grain boundary exhibits a classic pentagon-heptagon defect pattern, as well as highly distorted hexagons. Figure reproduced from paper by Huang et al. [74].

of dopant atoms also produces defect structures, but often with desirable changes in the electronic structure; this forms the basis of band gap engineering in graphene [76, 77]. In either case, careful structural analysis is needed in order to accurately model the effect of the defects.

Figure 8.4 shows an experimentally observed grain boundary, containing pentagon and heptagon defects, as well as highly distorted (defect) hexagons. Due to the image contrast provided by the electron density of the C-C bonds, the 'holes' in the middle of carbon rings are easily identified. For example, Ophus et al. [78] describe a method using local intensity filtering and peak detection to identify ring centres, from which the atomic coordinates are determined using a weighted Voronoi tessellation.

In a simulated graphene sample, only the atomic coordinates are available, from which the location of the holes is not immediately obvious. One possibility is to specify a maximum bond length, construct edges between all pairs of atoms which are less than this distance apart. Then, polygons can be found which are bounded by these edges. This approach suffers from the same drawbacks of CNA, in that it may not be possible to specify a consistent bond length in regions of high distortion. We describe a simple, parameter-free method for determining the vertices which bound each hole, from which the local geometric parameters can be determined in a similar manner to PTM.

Figure 8.5 shows a region of defective atoms in a graphene sheet and their Voronoi vertices. In the ordered hexagons, the Voronoi vertices are close together. With increasing distortion the spread in the Voronoi vertices also increases. This observation motivates a particularly simple method for identifying 'hole' polygons: atoms whose Voronoi vertices lie within a threshold distance belong in the same polygon. For the distance threshold, we use the bond length of graphene, 1.42Å. This choice of threshold is small enough to prevent neighbouring perfect hexagons from being merged, but large enough to prevent the aggressive subdivision of larger distorted polygons. The method

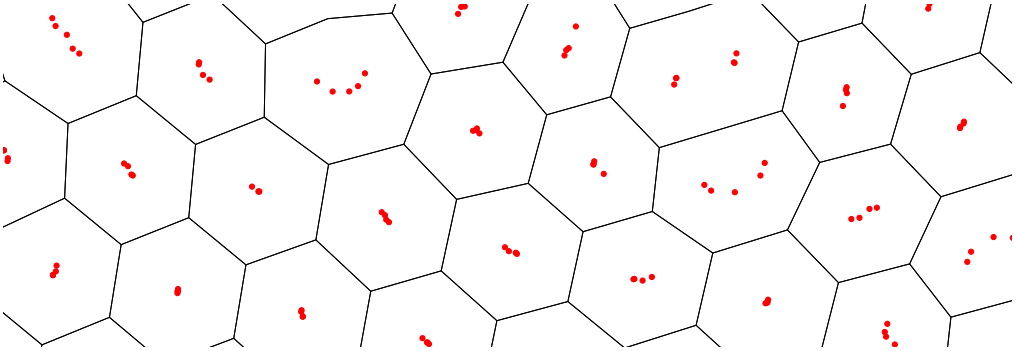


Figure 8.5. Voronoi-based construction of 'hole' polygons in simulated graphene sheets with structural defects. The Voronoi vertices of a regular polygon all lie at the polygon centre. Geometric distortion of the polygon spreads the Voronoi vertices out. By identifying atoms whose Voronoi vertices lie close together, a pleasing partition into polygons is obtained.

can be stated succinctly:

1	Calculate the Delaunay triangulation of the atom positions.
2	Calculate the circumcentre of each simplex in the Delaunay triangulation.
3	Iteratively group simplices whose circumcentres are less than threshold distance apart.
4	The exterior edges of polygon are those which appear only once in polygon simplices.

Figure 8.6 shows two simulated graphene sheets which have been analyzed in this way. The first graphene sheet is generated using the method of Ophus et al. [78], which aims to produce physically realistic grain boundaries with predominantly pentagon and heptagon defects, avoiding the larger interatomic distances and energetic penalties associated with higher-order polygons. The other graphene sheet† is generated by a MD simulation at a high annealing temperature, with the AIREBO potential of Stuart et al. [79]. Due to the many high-energy barriers which must be traversed in order to produce low energy interfaces, generation of polycrystalline graphene sheets with annealing produces many more higher-order polygon defects. This serves as a good test of the polygon identification method. In both data sets it can be seen that the resulting segmentation into polygons is visually pleasing, even in cases where the polygons contain many vertices, are severely distorted, or are even non-convex.

By assembling the exterior polygon edges into a path, the correspondence between atoms and template vertices is trivially determined. For example, in a well-formed hexagon, the vertices should be arranged in clockwise (or anticlockwise) order. With a correspondence determined in this way, the orientation, RMSD and strain can be calculated robustly, as is done in PTM. Calculating these properties using the

† Data kindly provided by J. Madsen.

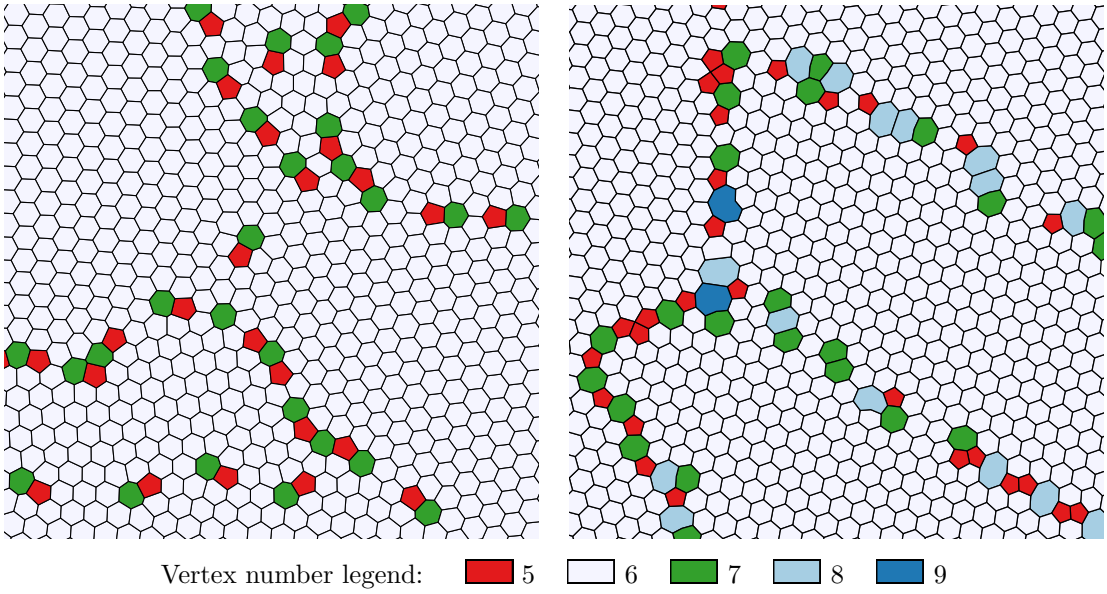


Figure 8.6. Identification of defect (non-hexagonal) polygons in polycrystalline graphene sheets. Atoms whose Voronoi vertices lie close together are collected into the same polygon.

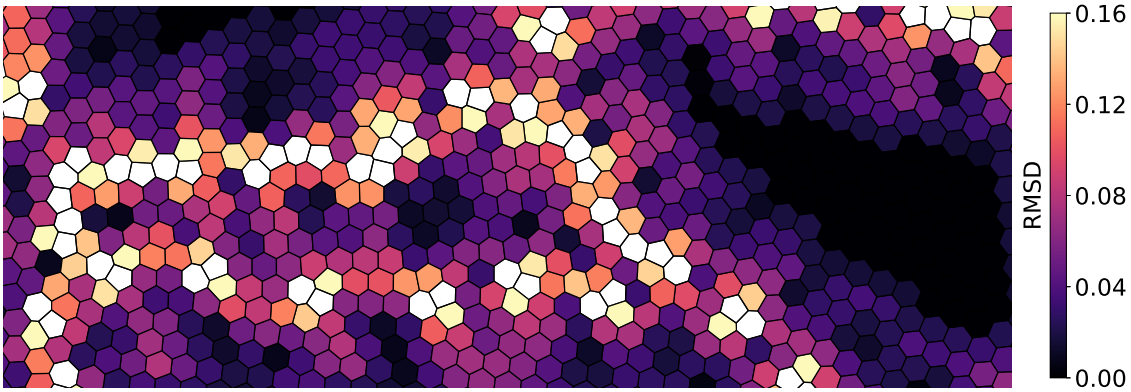


Figure 8.7. RMSD of hexagonal 'hole' polygons in a polycrystalline graphene sheet. The RMSD is calculated against all six atoms in a perfect hexagon template, which renders the calculation robust. As expected, geometric distortion is higher close to grain boundaries and other defects, and lower in the grain interiors.

coordinates of all six atoms of a hexagon represents an improvement to the methods in the literature [78, 80], typically used for calculating strains, where only two atoms (or lattice vectors) are used. The use of more local information renders the calculation more robust.

Figure 8.7 shows the RMSD of the hexagonal structures in a polycrystalline graphene sheet. Pentagons, heptagons, and other polygons are by definition defect structures; their RMSD is relatively uninteresting and has therefore not been calculated.

This method has been demonstrated for graphene, but could equally well be applied to hexagonal boron nitride, or other monolayer materials with little corrugation, i.e. materials whose atomic coordinates do not have a large out-of-plane component.

9 Interface Matching

In this section we will describe a method for matching interfaces between two crystalline thin films. Whilst the method is of general interest, it was developed for the specific application of studying interfaces between compound semiconductors and substrates. We will first introduce the better-studied topic of zero-strain homogeneous interfaces, before describing the method developed in Paper 2 which generalizes to heterogeneous interfaces.

9.1 Interfaces Between Lattices

In a polycrystalline sample, a grain is a contiguous, monophase crystallite, with a uniform crystal orientation. When two adjacent grains are of the same phase, the interface is known as a *grain boundary*. A well studied class of ordered grain boundaries are the so-called Coincidence Site Lattice (CSL) grain boundaries [81, 82, 83, 84], which, in combination with tilt and twist grain boundaries, form the basis of *grain boundary engineering* [85, 86], the production of stable materials with enhanced microtextural properties. Three examples of CSL grain boundaries are shown in Figure 9.1. Due to the sharing of sites, which results in a geometrically well-spaced distribution of atoms at the grain boundary, the interfaces have a low energy under a broad class of potentials.

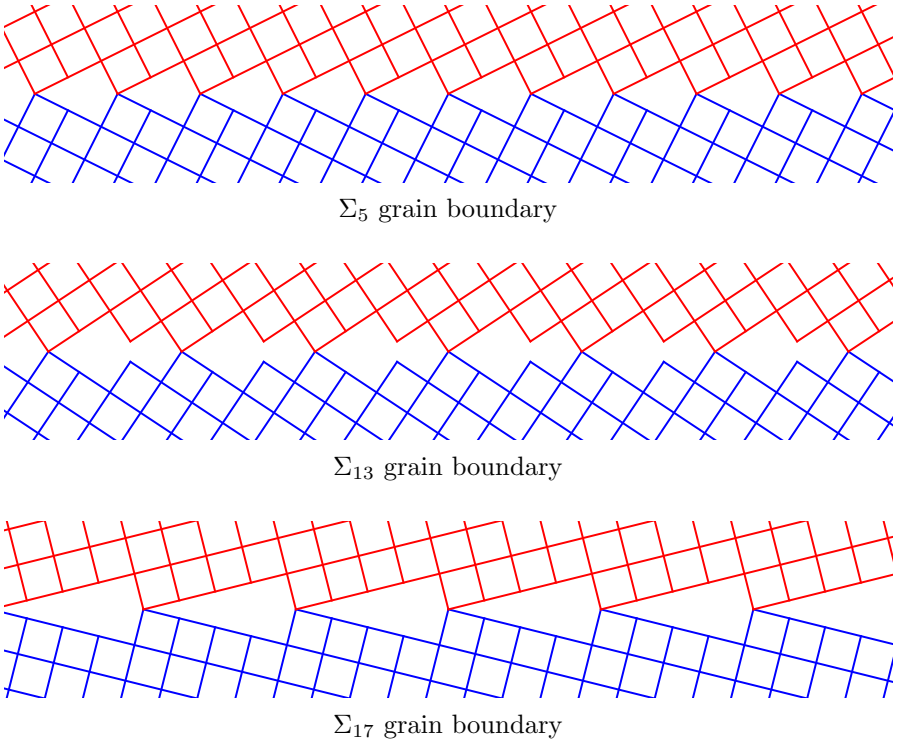


Figure 9.1. Three coincidence site lattice (CSL) grain boundaries between homogeneous square lattices. Per definition, a CSL interface has zero strain. Low-sigma CSL interface often have a low-energy due to sharing of sites. Interface matching in the general case can be considered a search for strained CSLs.

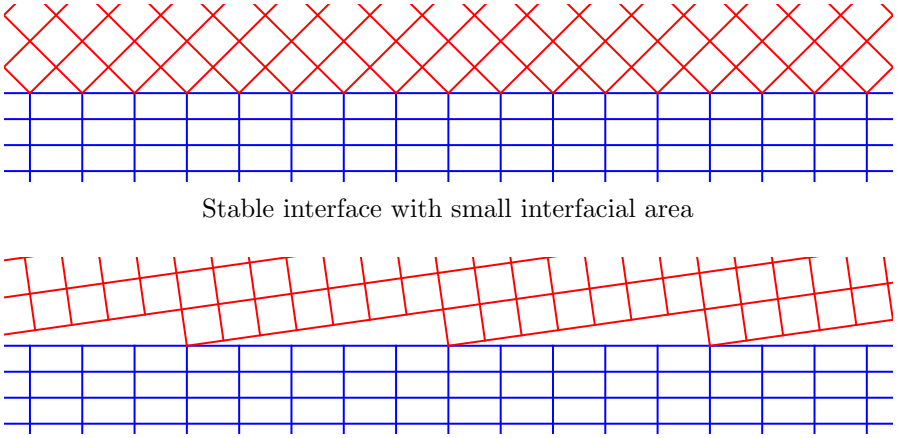


Figure 9.2. Unstepped and stepped interfaces between heterogeneous lattices. When an atomic basis is added to the lattices, the stepped interface has regions of both high and low atomic density, both of which incur an energetic penalty. As a consequence, structural relaxation is highly likely to result in a disordered interface.

Let us now consider an heterogeneous interface, by which we mean that the two sides of the interface contain different phases. A simple example is a Ag-Au interface. The lattice constants of Ag and Au, both in the FCC phase, are 4.085Å and 4.078Å, i.e. a difference of 0.175%. In a theoretical model, a zero-strain interface between any two crystalline materials can always be found, given that the lattice constants are not irrational. This observation has little practical relevance though, since the interfacial unit cells required for a zero-strain matching can be extremely large.

If we consider a more practical case, for example matching 111 planes of both Ag and Au, a small strain is required in order to make a stable interface. An interesting question is then: which ingredients are necessary for the formation of a stable interface? This is the central question posed by Stradi et al. [87]. Whilst the energetics of an interface clearly has a strong dependence upon the atomic basis of the crystal unit cells, they make two broad observations: an interface has a low energy if it has a low strain and a small interfacial unit cell.

The first observation has a simple explanation. The maintenance of high strain in a bulk material requires a large external force. Even at lower strains, strain is relieved via the production of dislocations [88, 89, 90], which in the thin film application studied here results in a disordered interface. The second observation is illustrated in Figure 9.2. A large interfacial unit cell results in a stepped interface. This results in both small and large interatomic distances, both of which incur an energetic penalty. An energetic model based on these observations, which does not include any atomic basis, therefore necessitates a purely geometric search for stable interfaces. In a homogeneous interface, this is essentially a search for a strained CSL interface.

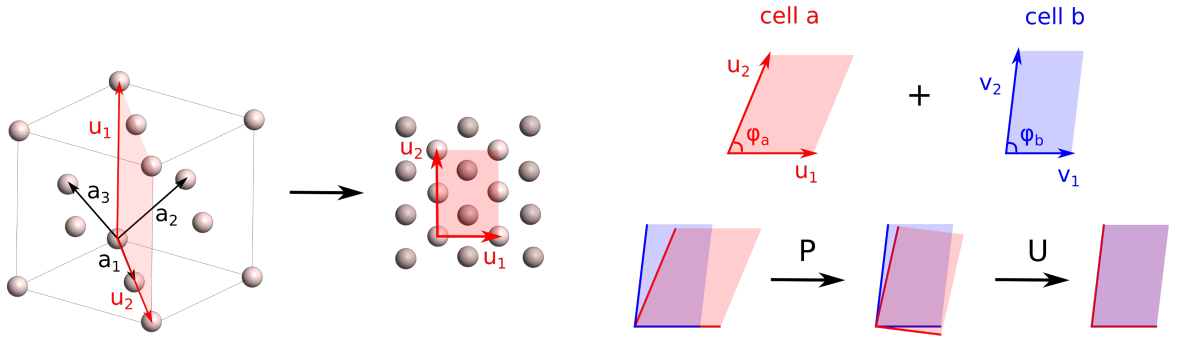


Figure 9.3. Illustration of the interface matching method, which consists of (1) selecting an interfacial plane in each lattice, and (2) finding the transformation which maps them onto each other to create an interface. **Left:** The interfacial plane in each lattice consists of two non-collinear vectors, each of which are integer combinations of the lattice vectors. **Right:** The interfacial cells are mapped onto each other with a linear map. The linear map is decomposed into a strain component **P** and a rotational component **U** using a polar decomposition. Figure reproduced from Paper 2.

9.2 Matching Algorithm

We have developed an algorithm which quickly identifies all possible low-strain interfaces between two lattices. The method is illustrated in Figure 9.3. Lattices are matched in an interface in a two-step process: (1) the interfacial planes of each lattice are specified, and (2) the planes are matched using a linear map. For each lattice, an interfacial plane is specified using two vectors which are integer multiples of the lattice vectors:

$$\vec{u}_1 = i_1\vec{a}_1 + j_1\vec{a}_2 + k_1\vec{a}_3 \quad \text{and} \quad \vec{u}_2 = i_2\vec{a}_1 + j_2\vec{a}_2 + k_2\vec{a}_3 \tag{9.1}$$

where $\vec{u}_1$ and $\vec{u}_2$ are the vectors which describe the plane, $\vec{a}_1$, $\vec{a}_2$ and $\vec{a}_3$ are the lattice vectors, and i, j and k are integers. A complete set of relevant planes can be built by iterating over i, j and k in the range $\{-m \dots m\}$, where $m \in \mathbb{N}^+$ denotes a cut-off beyond which the interfacial cells are expected to be too large to form stable interfaces.

Interfaces are created by pairwise combination of the interfacial planes from each lattice. For each pair of interfacial planes, we calculate the deformation gradient, $\mathbf{A}$, which maps the planes onto each other. Since the rotational component of the deformation gradient has no importance in the energetic model, a polar decomposition $\mathbf{A} = \mathbf{UP}$ is used to recover the strain, where $\mathbf{U}$ is the rotational component and $\mathbf{P}$ is the strain component. Whilst we consider cells here rather than sets of points, this method is identical to the non-rigid registration method described in section 3.2.

The method is demonstrated in Figure 9.4, where two semiconductor materials are matched against 10 different metals. In both materials, $InAs_{1-x}Sb_x$ and $Ga_xIn_{1-x}As$, the concentration, x, can be varied to allow the lattice constant to be very finely tuned. By tuning the lattice constant, the strain in the interface can be lowered, in some cases to zero. If we decompose a strain matrix $\mathbf{P} = \eta\mathbf{P}'$ into an isotropic hydrostatic component (a scaling factor), η, and a Von-Mises shear strain component $\mathbf{P}'$, then an optimal lattice constant tuning ensures that the hydrostatic component is 1.

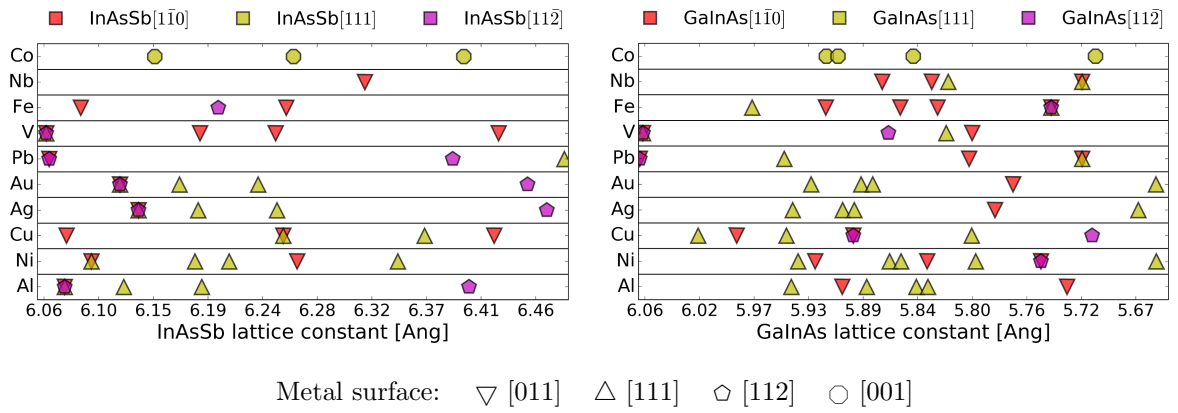


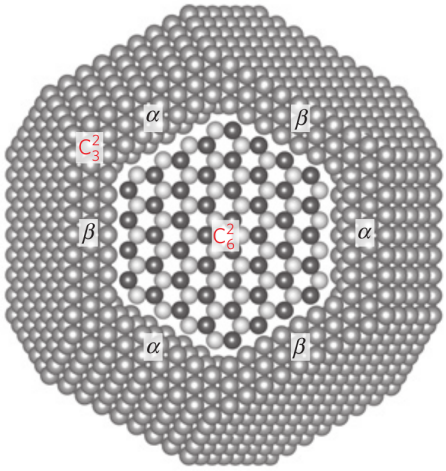
Figure 9.4. Perfect (zero-strain) matches between semiconductor and metal lattices. InAsSb (left) and GaInAs (right) are matched against ten different metals. The semiconductor surface is given by the colour (top legend); the metal surface is given by the shape (bottom legend). By varying the composition of the semiconductor, the lattice constant can be finely tuned, resulting in a zero-strain interface. Figure reproduced from Paper 2.

9.3 Motivating Application

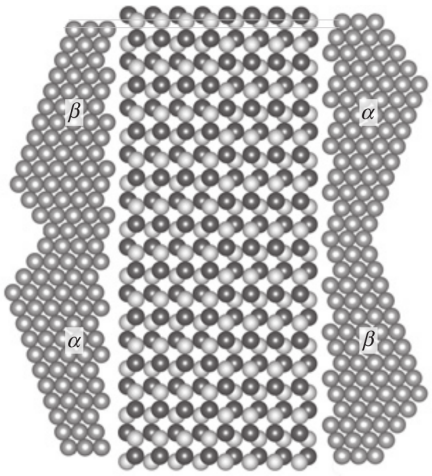
The application which motivates the study of low-strain interfaces is the production of core-shell nanowires. Nanowires with a semiconducting core and a superconducting shell have been shown to permit the realization of localized Majorana modes [91], which have applications in quantum information processing. Recently, it has been shown that disorder at the semiconductor/superconductor interface is responsible for decoherence in the Majorana modes [92], which prevents their use in quantum information processing. By the novel use of molecular beam epitaxy for growing the superconductor shell around the semiconductor core, a research group at the University of Copenhagen† has been able to construct nanowires with both the core and the shell in a crystalline phase [93]. The resulting interface is highly ordered and does not exhibit the Majorana mode decoherence which results from the use of conventional fabrication techniques.

Figure 9.5 shows a semiconductor/superconductor InAs-Al core-shell nanowire. The shell has been produced using molecular beam epitaxy, which results in a crystalline structure. The resulting interface is very well matched, and has a strain of 0.3%. Whilst the interfaces shown here are very low strain, they were found by experimentation. The method we present in Paper 2 replaces ad-hoc experimentation with a systematic investigation of all possible interfaces. Hopefully, this should enable the discovery of new semiconductor/superconductor combinations with enhanced properties.

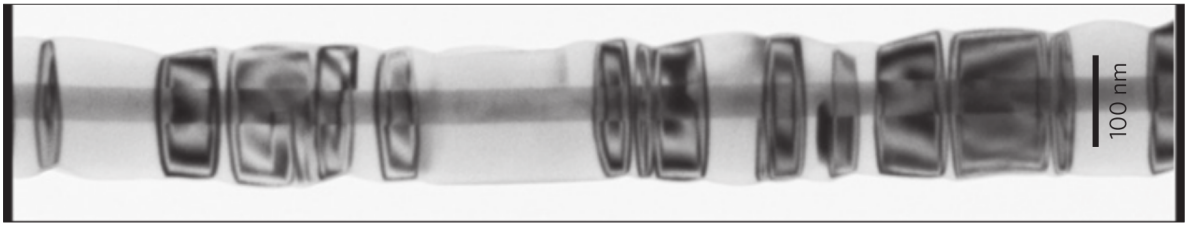
† J. Nygård's group at the Center for Quantum Devices. They also requested the development of a systematic method for finding low-strain interfaces, which motivated this work.



Top View



Side View



TEM Image

Figure 9.5. Nanowire with an InAs core surrounded by a polycrystalline Al shell. **Top:** Top and side views of the nanowire. Both views illustrate the crystalline order of the superconducting shell, as well as the good match at the interface to the semiconducting core (0.3% strain). At any point along the wire's length, the shell consists of six grains, each of which are in one of two orientations, denoted α and β. **Bottom:** A TEM image of the nanowire. The difference in contrast in the shell is due to two different cyclic orders of the shell grains, either $\alpha\beta\alpha\beta\alpha\beta$ or $\beta\alpha\beta\alpha\beta\alpha$. Figures reproduced from the paper by Krogstrup et al. [93].

10 Cluster Expansion Models

Determination of the structure of an alloy via experimental measurements has a significant advantage over computer simulations, in that the ground state configuration is immediately revealed. The obvious drawback is that the time required to perform an experimental measurement limits the number of systems which can be investigated.

Given a chosen alloy composition, a computer simulation must search over both the geometric structure and the chemical ordering. This is an enormous configurational space to investigate. Often, though, the geometric structure is known a priori. For example, it may be known that an alloy prefers to form FCC lattices, or, if nanoparticles are being studied, it may be possible to fix the topology. With the geometric structure fixed, the problem is reduced to that of finding an optimal chemical ordering. Although the space of chemical orderings scales exponentially with the number of atoms considered, this represents an enormous reduction from the full configurational space, and in turn permits a simplified energetic expression.

Evaluation of a full Hamiltonian is expensive, both for semi-empirical potentials and particularly for ab initio methods. A full Hamiltonian must handle all possible atomic positions and chemical orderings, a space which in principle encompasses an infinite range of energies. With a fixed topology, the range of energies is significantly reduced, and the Hamiltonian can be distilled into a so-called 'model Hamiltonian'[94] via the use of a cluster expansion (CE) model. A CE model consists of weighted multi-site correlation terms (c.f. Figure 10.1), with products of pseudo-spin variables $\sigma_i \in \{-1, 1\}$ at each site determining the energetic contributions:

$$\begin{aligned} H(\boldsymbol{\sigma}) = V_0 &+ \sum_{\mathbf{c}_f \in \mathbf{C}_1} \sum_{i \in \mathbf{c}_f} V_{1,f} \sigma_i \\ &+ \sum_{\mathbf{c}_f \in \mathbf{C}_2} \sum_{(i,j) \in \mathbf{c}_f} V_{2,f} \sigma_i \sigma_j \\ &+ \sum_{\mathbf{c}_f \in \mathbf{C}_3} \sum_{(i,j,k) \in \mathbf{c}_f} V_{3,f} \sigma_i \sigma_j \sigma_k + \dots \end{aligned} \tag{10.1}$$

where $H(\boldsymbol{\sigma})$ is the energy of a configuration $\boldsymbol{\sigma}$, $\mathbf{C}_n$ is the set of all n-body correlation terms (the clusters), each containing cluster instances $\mathbf{c}_f$, and $V_{n,f}$ is the effective cluster

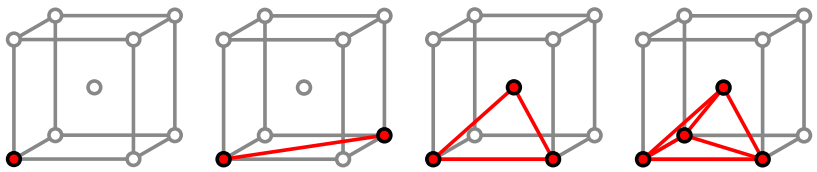


Figure 10.1. Examples of multi-site correlation clusters in a BCC lattice for n in the range 1 to 4. In a cluster expansion model, each correlation term is the product over the spin variables in a cluster.

interaction (ECI) for the f^{th} n-body cluster instance.

In their seminal article on CE models [95], Sanchez et al. show that the clusters form an orthogonal basis. If the basis is complete, the model represents any Hamitonian exactly, i.e. a full set of clusters implies perfect representation of physical properties. For anything but the smallest number of sites a full model is impractical; with n sites there are 2^n clusters in total. As such, the model is truncated. Even with truncation, a CE model is capable of good accuracy; since most physical interactions are predominantly short-ranged, it is expected that low-order and short ranged clusters have higher magnitude coefficients, and higher-order and very long ranged clusters have insignificant contributions.

Parallels can be found with other expansions into orthogonal bases; in the expansion of a natural sound in a truncated Fourier basis, it is expected that the fundamental frequency will have a large (if not the largest) absolute coefficient. In modern image and video compression methods [96, 97], the discrete cosine transform is used as a basis; under the prior of natural image statistics (locally correlated pixel intensities, scale invariant power spectrum) this is shown to concentrate the coefficients with the largest magnitude in the low-order terms. Indeed, a CE model can be considered as a compressed form of a full Hamiltonian under a prior of a fixed topology and chemical content.

Use of a CE model involves four steps:

(1) Enumeration of potential clusters for inclusion in the CE model.

(2) Selection of relevant clusters from the pool of possible clusters.

(3) Construction of a training data set, and optionally a separate test data set.

(4) Determination of the ground state structures using the fitted CE model.

The selection of data points for test and training is well discussed in the literature; methods include using independent and identically distributed random variables for structure generation [98], iteratively fitting a CE model to lower-energy structures [99, 100], and including high-energy structures in order to improve generalization [101]. In the rest of this section, we will discuss points 1, 2, and 4, to which we make novel contributions.

10.1 Enumeration of Potential Clusters

Whilst CE models have seen extensive use, primarily in periodic crystalline systems and surfaces of crystalline systems, the widespread application to non-crystalline systems such as nanoparticles appears to be hindered by some practical difficulties. There exist three widely-used frameworks for cluster expansions: UNCLE [102], ATAT [103, 104], and TTK [105, 106]. Each of these performs automatic cluster identification and selection, saving the user from doing this by hand. These codes, however, appear to support only crystalline systems (i.e. lattices).

CE models have been used to model nanoparticles (which are by definition not periodic lattices). Tan et al. [99] have for example studied FCC nanoparticles, and enumerated clusters by modifying the periodic lattice method in the TTK framework. More recently, Teeriniemi et al. [107] have used the coordination numbers of each atom to enumerate clusters in decahedral and icosahedral nanoparticles. Each of these methods exploits the specific geometry of the system to be studied. Here, we present a method based on geometric graph theory to identify all geometrically unique clusters in any system, which relies only on the interatomic distances and (optionally) site types.

Given a set of n sites, we wish to group all geometrically equivalent subsets. We can do so in two steps. First, for each subset of size k, we start by identifying all combinations of k sites. This is the easy part, as it requires a simple enumeration of combinations. The second step is to group equivalent sites, which we do by finding the canonical form of each subset. We must define a representation such that subsets with the same canoncial form are geometrically equivalent.

For a subset of k sites, we construct a distance matrix of size $k \times (n+1)$, where each row contains the distance of site i to every other site. If the structure contains different sites, the first element of the row is the site type. This subset distance matrix, $\mathbf{M}$, is given by:

$$\mathbf{M}(P) = \begin{bmatrix} s_{P(1)} & d_{P(1),P(1)} & d_{P(1),P(2)} & d_{P(1),P(3)} & \dots & d_{P(1),P(n)} \\ s_{P(2)} & d_{P(2),P(1)} & d_{P(2),P(2)} & d_{P(2),P(3)} & \dots & d_{P(2),P(n)} \\ s_{P(3)} & d_{P(3),P(1)} & d_{P(3),P(2)} & d_{P(3),P(3)} & \dots & d_{P(3),P(n)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ s_{P(k)} & d_{P(k),P(1)} & d_{P(k),P(2)} & d_{P(k),P(3)} & \dots & d_{P(k),P(n)} \end{bmatrix} \tag{10.2}$$

where s_i is the site type of site i, $d_{i,j}$ is the distance between sites i and j, and P is a permutation. The canonical form, $C(\mathbf{M})$ is given by the permutation which minimizes the lexicographic order:

$$C(\mathbf{M}) = \arg\min_{P} \mathbf{M}(P) \tag{10.3}$$

By using this method, all geometrically unique clusters are found in a systematic manner. No knowledge of the point group of the system is required, since any rotational, chiral, and/or translational symmetries are automatically detected and do not require explicit inclusion. We will now demonstrate this cluster enumeration method in a nanoparticle system.

Figure 10.2 shows a 147 atom icosahedral nanoparticle, and the number of unique clusters at every spatial extent. Due to the finite size of the nanoparticle, the maximum spatial extent of a cluster is severely limited. On the whole though, the number of unique clusters increases with increasing spatial extent, and dramatically so with the number of cluster sites. In the above example, when including cluster bodies of size $n = 1, 2, 3, 4$, the total number of clusters is over half a million. A CE model with more than 100 clusters is considered large; thus, selection of a small cluster basis from a huge number of potential clusters poses a significant challenge. This discussion brings us onto our next section.

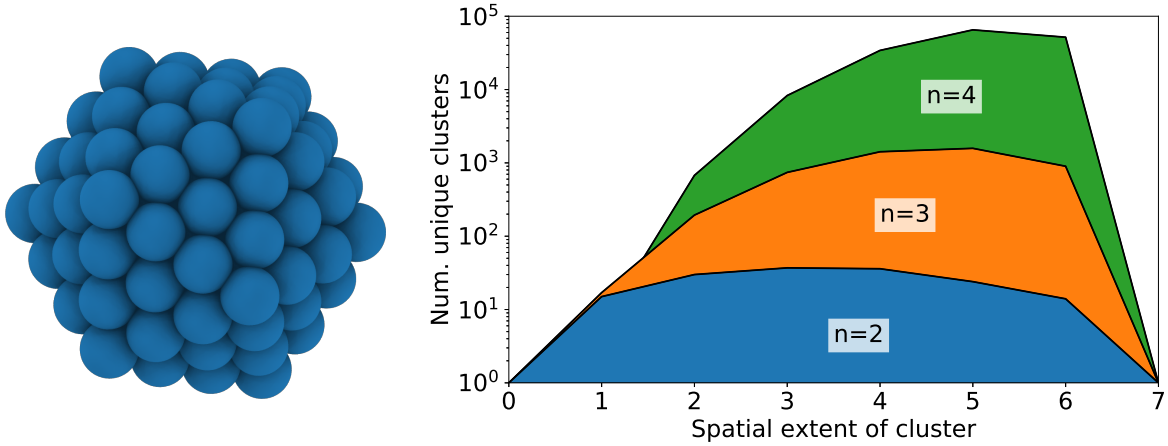


Figure 10.2. Left: A 147 atom 'magic' icosahedral nanoparticle. **Right:** Number of geometrically unique clusters at each spatial extent. For simplicity, the spatial extent shown here is the maximum distance between sites in a cluster, where the distance is given by the number of 'hops' between nearest neighbours.

10.2 Cluster Selection

As we have stated above, a CE model consists of ECIs and clusters. If a set of clusters has been chosen, the methods for selecting the ECIs are well established. The error term to be minimized is typically the mean absolute error (MAE) or the root-mean-square-error (RMSE). If the set of clusters is fixed, L2 regularization (also known as ridge regression, or Tikhonov regularization [108]) may be applied in order to improve predictive performance. The question of how to choose clusters, though, is still rapidly evolving.

The choice of clusters is critical to the performance of a CE model. In the seminal paper on cluster expansions by Sanchez et al. [95], the authors show that a cluster expansion is an orthogonal basis, but do not prescribe a cluster selection strategy other than stating that the basis must be truncated. A physically inspired strategy, conceived by Zarkevich and Johnson, is to construct a self-consistent cluster basis using two-rules [109]:

- 'If an n-body cluster is included, then include all n-body clusters of smaller spatial extent'
- 'If a cluster is included, include all its subclusters'

With these selection rules, the parameters which determine the cluster basis are the maximum cluster size (n), and the spatial extent.

Later work ignores the self-consistency principle, and chooses clusters according to their predictive ability alone, rather than trying to assign any physical meaning to the clusters and associated ECIs. Methods for 'predictive' cluster selection are predominantly based on either genetic algorithms [101], or compressive sensing methods [98, 110] based on L1 minimization. The latter method has highly interesting properties and warrants further discussion.

In the solution of a linear system of equations, regularization imposes constraints on the solution vector. Consider, for example, a least-squares problem with features A, observations b, and variables x:

$$||Ax - b||_2^2 \quad \text{Subject to} \quad ||x||_2^2 < \kappa \tag{10.4}$$

Here, the solution vector is constrained to lie within a hypersphere of radius κ^2. In Lagrangian form, the constraint is soft, i.e. the norm of the solution vector is penalized rather than being constrained to a maximum size:

$$||Ax - b||_2^2 + \lambda ||x||_2^2 \tag{10.5}$$

This is known as L_2 regularization. Here, the magnitude of the variables is reduced without changing the choice of features. A relatively recent development [111] is the use of the L_1 norm for regularization. The L_1 norm of a vector is the sum of the magnitudes of each vector component, also called the taxicab or Mahattan distance. We can constrain the solution vector using this norm:

$$||Ax - b||_2^2 \quad \text{Subject to} \quad ||x||_1 < \kappa \tag{10.6}$$

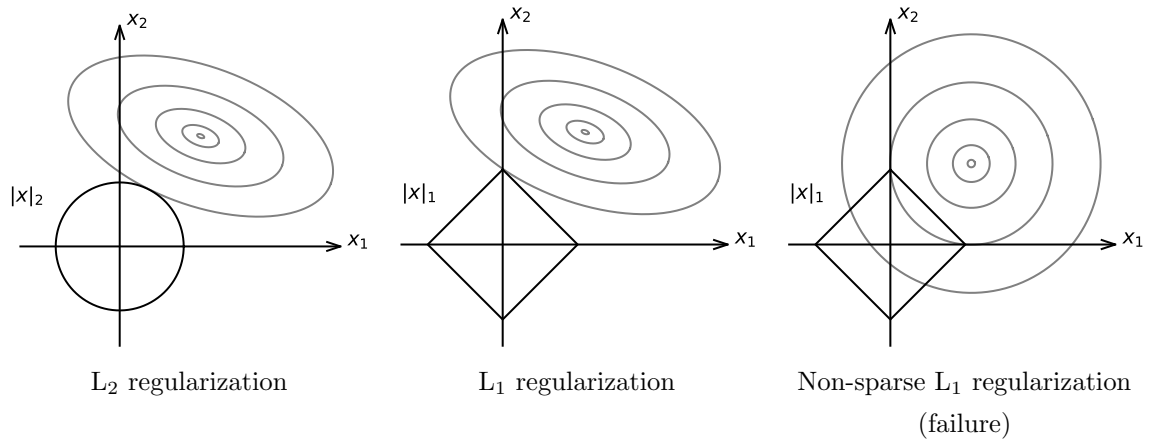
in which case the solution vector is constrained to lie within a cross polytope. As in L_2 case, the Lagrangian form penalizes the norm of the solution vector:

$$||Ax - b||_2^2 + \lambda ||x||_1 \tag{10.7}$$

Regularization with the L_1 norm has a remarkable property, in that it (in many cases) induces sparsity on the solution vector. A geometric illustration of this property is given in Figure 10.3. What this means in a cluster expansion context is that, by changing the regularization parameter, the size of the cluster basis can be varied. In addition to the cluster selection, the ECIs are simultaneously fitted. This approach has seen successful application in a range of CE studies [98, 112], but, as we shall demonstrate, it does not always represent the best choice for cluster selection.

The best possible method for sparse feature selection is L_0 'regularization'. The L_0 'norm' of a vector is the number of non-zero values it contains. It is not a true norm, since it does not fulfil the criteria of a metric, but it is a useful measure of sparsity. The problem of feature selection the L_0 norm can be stated: given n features (columns in the A matrix), find the subset of size k which globally minimizes the objective function. The success of the L_1 norm is due to it being a good approximation of the L_0 norm. However, a good approximation is dependent on the satisfaction of specific conditions of the A matrix, which are difficult to verify (for a good overview, see the paper by Bertsimas et al. [113]). In the presence of highly correlated features or noisy data, the sparsity induced by the L_1 norm is diminished, and the predictive capacity of a model built in this way suffers.

More fundamentally though, the L_1 norm cannot improve upon the L_0 norm, for the simple reason that the latter is *by definition* the best possible solution. This being the



L_2 regularization | L_1 regularization | Non-sparse L_1 regularization (failure)

Figure 10.3. Geometric illustration of the L_1 and L_2 regularization methods. **Left** The L_2 norm is isotropic. Whilst penalization of the L_2 solution norm can often improve predictive performance, it does not produce a sparse solution. **Centre** The L_1 norm is not isotropic. The extreme values of many functions lie at the corners when the solution vectors is constrained under the L_1 norm. Since the corners lie along the solution axes, a sparse solution is produced. **Right** Construction of a failure case is trivial. Here, no amount of L_1 regularization produces a sparse solution.

case, why has the L_0 norm not seen any use in the context of CE models? The answer is that, whereas efficient solvers exist for L_1 minimization (even in polynomial time), solution of the L_0 norm is NP-hard [114]. For small CE models though, we can use Mixed Integer Programming (MIP) or Mixed Integer Quadratic Programming (MIQP) to find optimal cluster bases with the L_0 norm (c.f. Paper 4 for an introduction to MIP modelling). Model 10.1 shows a mathematical model for the cluster selection problem using L_0 norm. Using this model, we can describe a recipe for the construction of a CE model:

(1) Identify all clusters using the graph theoretic method described previously.

(2) Select an optimal cluster basis of size k using a L_0 regularized MIP/MIQP model.

We emphasize that no reduction of the cluster basis should be done prior to L_0 optimization. For example, reduction of a matrix to linearly independent columns should be avoided, as illustrated in Figure 10.4. Using this recipe, we have built a cluster model for investigating stable alloy structures for use in battery materials.

10.3 Application to a Lithium Ion Battery Material

Here, we apply the methods described previously to build a CE model for a novel lithium ion battery alloy †. Lithium ion batteries provide a much lower energy density than e.g. fossil fuels. The demand for increased storage capacity requires new and improved alloy structures. It has recently been discovered that a disordered rock salt

† This is a research project of Tejs Vegge's group at DTU Energy. I thank J.H. Chang, S. Loftager, J.M. Garcia, and Tejs, for many discussions about their research and for kindly providing me with their data.

Indices:	$i \in \{1, \ldots, m\}$		Row/observation index	(10.8)
	$j \in \{1, \ldots, n\}$		Column/cluster index	(10.9)
Variables:	$\epsilon_i \in \mathbb{R}$	$\forall i$	Absolute prediction error i	(10.10)
	$x_0 \in \mathbb{R}$		Intercept	(10.11)
	$x_j \in \mathbb{R}$	$\forall j$	ECI of cluster j	(10.12)
	$y_j \in \{0, 1\}$	$\forall j$	Cluster j selection variable	(10.13)
Parameters:	$\mathbf{A} \in \mathbb{R}^{m \times n}$		Cluster instance counts	(10.14)
	$\mathbf{b} \in \mathbb{R}^m$		Observations	(10.15)
	$n_\mathrm{c} \in \mathbb{N}$		Desired number of clusters	(10.16)
	$x_\mathrm{max} \in \mathbb{R}$		Max. absolute ECI value	(10.17)
Minimize:	$\frac{1}{m} \sum_i \epsilon_i$		Mean absolute error	(10.18)
(alternatively)	$\frac{1}{m} \sum_i \epsilon_i^2$		Mean squared error	(10.19)
Subject to:	$\sum_j y_j = n_\mathrm{c}$		Fixed number of clusters	(10.20)
	$x_j \leq x_\mathrm{max} y_j$	$\forall j$	ECI=0 if cluster not selected	(10.21)
	$x_j \geq -x_\mathrm{max} y_j$	$\forall j$	ECI=0 if cluster not selected	(10.22)
	$\epsilon_i \geq x_0 + \sum_j A_{ij} x_j - b_j$	$\forall i$	Absolute error constraint	(10.23)
	$\epsilon_i \geq -x_0 - \sum_j A_{ij} x_j + b_j$	$\forall i$	Absolute error constraint	(10.24)

Model 10.1. A [MIP/MIQP] model for minimizing the [mean absolute error (MAE)/mean squared error (MSE)] of a cluster expansion model, with L_0 regularization.

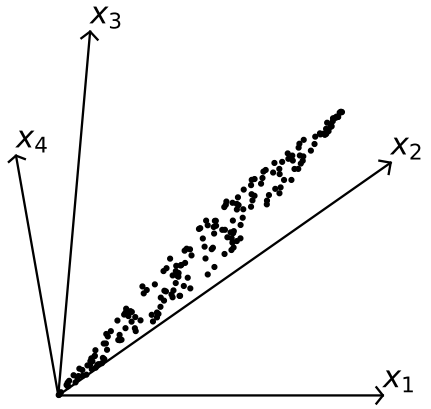


Figure 10.4. Geometric illustration of low-rank feature selection. Since the data has rank 2, any two features from (x_1, x_2, x_3, x_4) can describe the data exactly. When a sparse (low-rank) model is required, the choice of features is significant. Here, x_2 is clearly the best feature, as it describes more of the variation of the data than the other features.

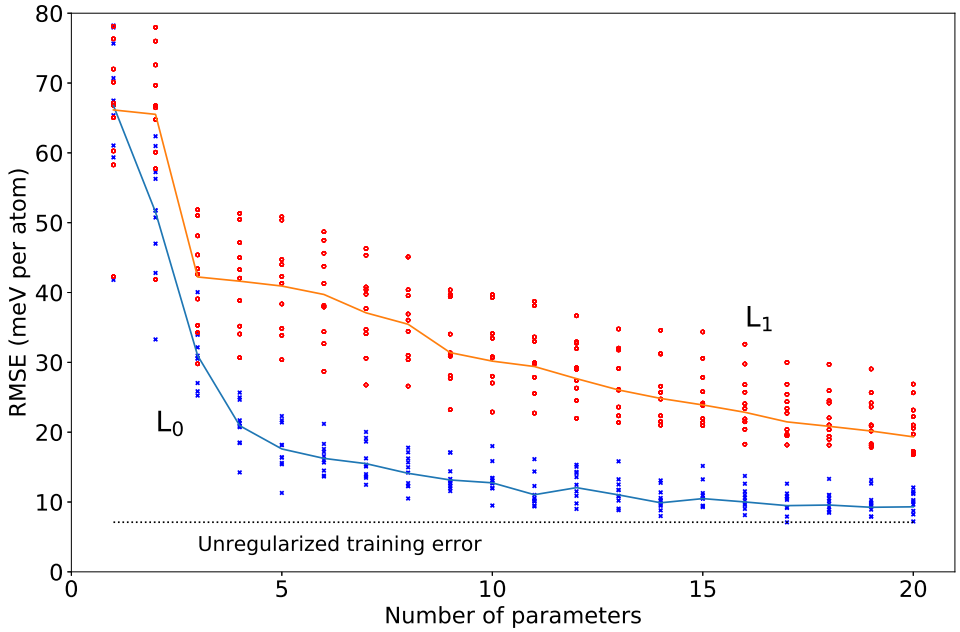


Figure 10.5. Comparison of L_0 and L_1 regularization, using 10-fold cross validation. The mean training error for each number of parameters is plotted as a line (the first parameter is the intercept). The scattered data points are the errors for each of the 10 test data sets. L_0 regularization performs significantly better at every number of parameters, both for the training and test sets. Importantly, the variance in the test errors is much smaller.

(DRS) structure has improved lithium ion diffusion. The improvement, though, comes at a cost of a reduced lifetime. Structural defects accumulate during charge-discharge cycling, and after a few dozen cycles the battery capacity is significantly degraded. Prevention of the structural degradation motivates the search for new, highly stable structures.

One such promising structure is a Lithium/Vanadium/Oxygen/Fluorine alloy. For a good overview, we refer to papers by Chen et al. [115], Cambaz et al. [116], and Ren et al. [117], on which the study of this alloy is based. As stated above, the alloy forms a DRS (disordered NaCl) structure, where Li and V atoms occupy the Na sites, and O and F atoms occupy the Cl sites. Using ab initio methods, the energies of 217 18-atom structures, and 67 54-atom structures have been calculated. Each structure is symmetrically and energetically unique, and has periodic boundary conditions applied (to simulate a bulk crystalline structure).

This system provides an excellent test of the methods presented here; it has practical relevance, two site types, multiple elements, and is of a tractable size. We begin by enumerating clusters. We choose a maximum cluster size of $n = 4$. Due to the small size of the structure, there is no need to select a maximum spatial extent of the clusters, as this is instead limited by the cells periodicity.

From the enumerated set of clusters, we select provably optimal subsets of different sizes by solving the L_0 norm problem shown in Model 10.1. To do so, we use the

MIQP software framework of Bertsimas et al. [113] in combination with the Gurobi solver [118]. The results are shown in Figure 10.5. It can be seen that the L_0 solution is significantly better at all subset sizes. More importantly, the solution quality converges much faster. For example, visual extrapolation would suggest that the L_0 solution with ≈ 12 parameters would outperform an L_1 solution with more than double the number of parameters. Significantly, the reduction in test error variance highlights the benefits of sparsity when building a CE model.

10.4 Ground-State Solution of a CE Model

We have now described methods for systematic identification of geometrically unique clusters and construction of a CE model. The last step is finding the ground state of the CE model. With the exception of a recent paper by Huang et al. [119], where a provable ground state is found using *weighted MAX-SAT* [120], every other study of CE models in the literature uses stochastic search to attempt to find a ground state. Whilst stochastic methods are useful for obtaining a ensemble of configurations at a specified temperature (e.g. room temperature), they cannot guarantee that the true 0K configuration is found.

In Papers 3 and 4 we have used MIP models to find provable ground states of CE models in a selection of nanoparticle systems. We refer to the papers for details, but briefly discuss one of the highlights here. Figure 10.6 shows ground state chemical ordering in a 309-atom nanoparticle with 'magic' icosahedral morphology, at a selection of concentrations. The CE model captures local relaxations, but does not otherwise permit large changes in the nanoparticle morphology. Even so, the total number of configurations, 2^{309}, is larger than can be explored exhaustively. By using a MIP model, the ground states can be found in a few minutes to a few hours. The nanoparticle ground states have surprisingly complex orderings; they exhibit neither phase-segregration nor phase mixing, but a complex interplay between surface and mixing effects.

10.5 Summary

In this section, we have demonstrated a systematic approach for cluster construction, and optimal solutions for truncated cluster basis selection. The exact determination of ground states in a CE model forms the basis of Papers 3 and 4, where ground states are found in nanoparticle systems†. Paper 5 solves, the inverse problem of a CE model. Rather than finding the ground state of a single, compex CE model, we find all possible ground states of simple CE models.

† Due to some specific EU project funding conditions, I am unfortunately unable to present the ground states for the lithium ion battery application discussed above. Readers may choose to accept my assurances that this is an easy MIP case to solve.

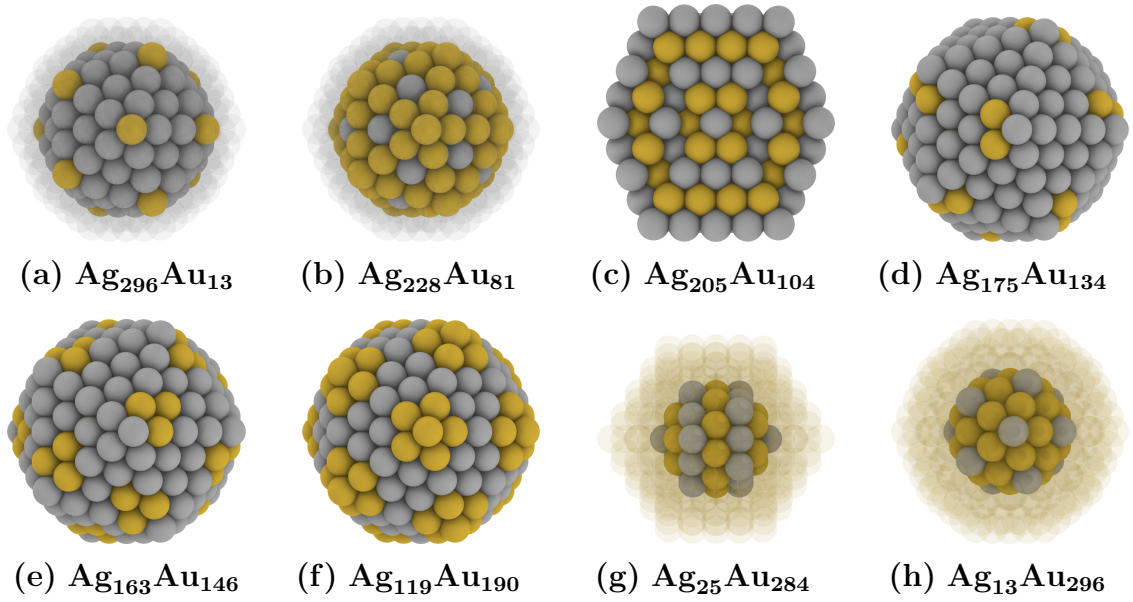


Figure 10.6. A selection of ground-state configurations in a 309-atom Mackay icosahedron AgAu nanoparticle. The configurations exhibit a diverse range of highly ordered interior and surface structures. In **(a)** and **(b)**, the transparent atoms represent a single layer of Ag atoms; in **(g)** and **(h)** they represent two layers of Au atoms. The Au atoms exhibit a strong preference for subsurface shell sites, in particular the subsurface corners **(a)**, followed by the subsurface edges **(b)**. The Ag atoms show a strong preference for 2^{nd} shell corner sites **(g)**-**(h)**, as well as the central atom. The lowest energy configuration **(c)** (shown as a slice through the nanoparticle) has a perfectly ordered onion-shell structure. At high Au concentrations, the preference of Au atoms for corner, edge or interior sites of the surface facets varies as a function of composition **(d)**-**(f)**. Figure and caption reproduced from Paper 3.

11 Pattern Indexing

In this section we describe the experimental determination of material properties by analysis of diffraction patterns. Indexing of diffraction patterns is the motivation behind Papers 6 and 7. In the context of this thesis, we again apply the *holy trinity* of computational geometry (the convex hull, the Delaunay triangulation, and the Voronoi diagram) to generate nearly-optimal sets of orientations for use in the indexing of diffraction patterns. We also identify a case in an existing pattern indexing approach where the use of the convex hull leads to poor performance.

11.1 Indexing of Diffraction Patterns

Theoretical or computational studies of materials have many useful applications. For example, when searching for a new material which has some desired property, stable materials can be quickly screened using ab initio energy calculations. Another example is the modelling of dislocations using MD simulations, which permits a better understanding of deformation modes by providing fine-grained snapshots at the atomic-level.

Nonetheless, experimental measurements are of critical importance. Any computational model is fitted to experimental results; the semi-empirical potentials typically used in MD simulations are fitted to known experimental values such as the energies of sublimation and vacancy-formation, lattice constants, and elastic constants [121, 122]. Even in some 'ab initio' methods, experimental data are used; some exchange and correlation functionals used in DFT are fitted parameters [123, 124].

One of the most fundamental experimental techniques is the study of crystalline structures using diffraction. A veritable cornucopia of modern diffraction modalities exists for neutron, X-ray, and electron sources. By collecting images of the diffraction pattern of a sample, its crystal structure can be determined. The diffraction pattern of a sample is a projection of the reciprocal lattice of the sample, from the 3D reciprocal space of the lattice to the 2D space of the detector. Despite the loss of information which

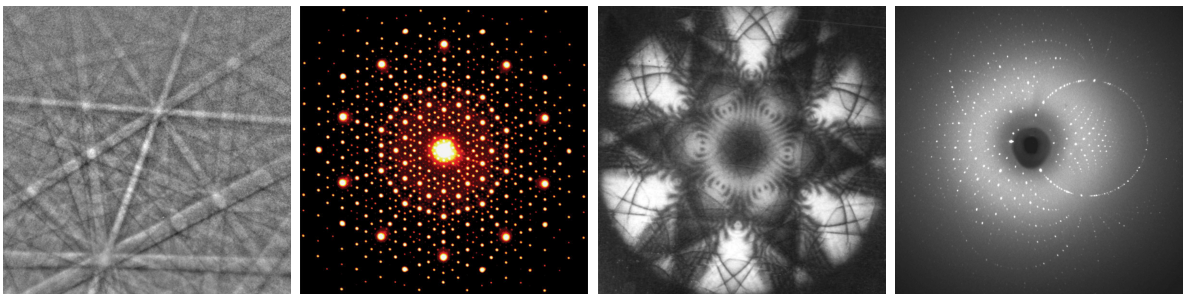


Figure 11.1. Diffraction patterns from a selection of modalities: Electron Backscatter Diffraction, Transmission Electron Microscopy, Large Angle Convergent Beam Electron Diffraction, and X-ray synchrotron diffraction. Images reproduced from papers [125, 126, 127, 128].

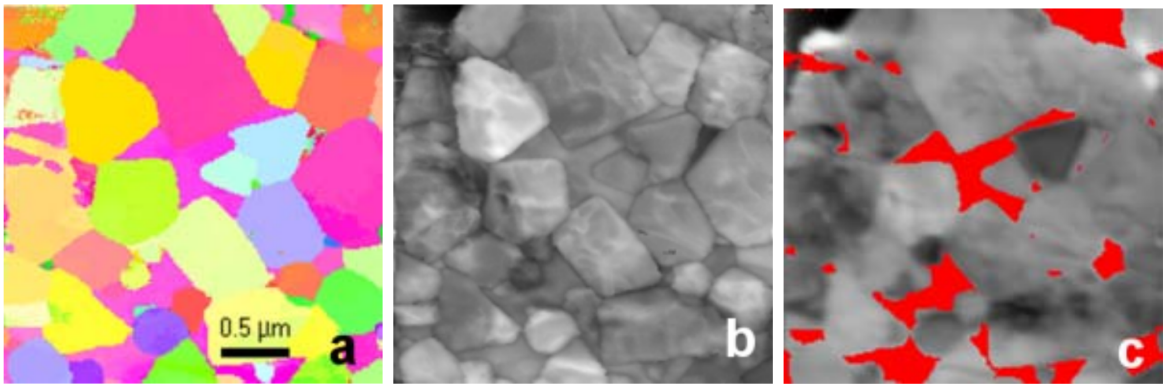


Figure 11.2. Orientation, correlation index, and phase maps in a two-phase (Co-WC) polycrystalline sample, obtained using TEM diffraction. **(a)** The orientation map is a false-colour representation to highlight the different grain orientations. **(b)** The correlation index is correlation between the predicted scattering vector intensities (for the best orientation) and the experimentally observed intensities. Defects (such as grain boundaries, dislocations, and point defects) are by definition not crystalline, and show up as dark regions in the correlation index map. **(c)** The Co phase is marked in red. Images reproduced from paper by Rauch et al. [129]

the projection entails, careful analysis of the geometry and intensities of the diffraction pattern can (in most cases) reveal the crystalline structure to a high degree of certainty. Figure 11.1 shows a selection of diffraction images.

In addition to the crystalline structure, pattern indexing can be used to determine variety of physical properties such as the strain and orientation of the diffracting volume, as well as the presence of defects, as shown in Figure 11.2. We will focus here on the determination of the crystal orientation. Given that the crystal structure (unit cell and atomic basis) has been determined, the crystal orientation can be determined in one of essentially two ways: solution of the *inverse problem*, and *forward modelling*. An inverse problem, in a more general context, is the determination of the underlying configuration from the observations. Forward modelling goes in the opposite direction: observations are predicted by selecting an underlying configuration. By searching over a subset of all possible underlying configurations, the best estimate is selected as the one with the most overlap between the predicted and experimental observations.

Solution of the inverse problem is the older of the two approaches, due to the smaller computational requirements. For example, in Electron Backscatter Diffraction (EBSD), the Hough transform [130, 131] is used to identify Kikuchi lines in the diffraction pattern. This approach permits the indexing of hundreds of patterns per second. Inverse approaches, however, fail in the presence of noise. In, for example, a noisy EBSD pattern, the Hough transform cannot reliably find the Kikuchi lines, and the indexing procedure fails as a consequence. This limitation motivated the use of forward modelling.

Development of forward modelling methods started in the 90's [132], but it took almost 25 years until the necessary computational power was sufficient [133], even requiring the use of supercomputers [134], until the recent development of GPU-based

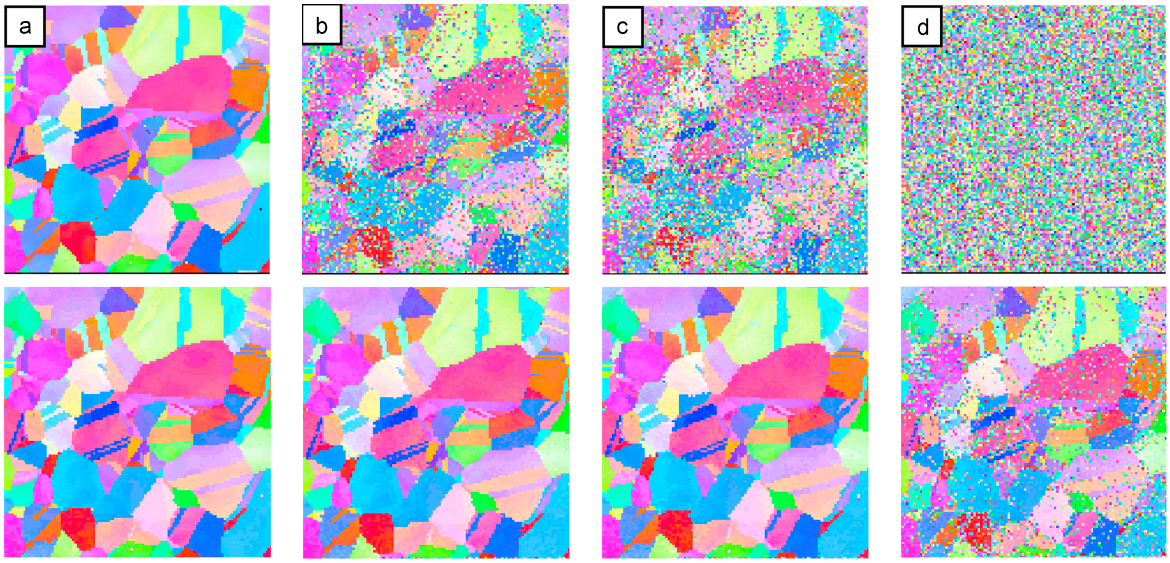


Figure 11.3. Comparison of EBSD pattern indexing methods at different noise levels for a polycrystalline nickel sample. The Hough transform based indexing (top) performs significantly worse than the dictionary-based forward modelling indexing in the presence of noise. Gain levels are (a) 0dB (b) 29.3dB (c) 30.4dB and (d) 36.0dB. Figure reproduced from the paper by Wright et al. [137].

algorithms enabled practical use on desktop computers [135, 136]. As mentioned above, forward modelling selects the best orientation by searching over every orientation in a discretization of $SO(3)$ (the space of orientations). In modern methods, a predicted diffraction pattern is generated for each orientation using a physics-based simulation of the diffraction processes, as well as the detector response. By comparing the predicted patterns to the experimentally observed pattern, the best orientation can be identified. Despite the inability of most physical models to perfectly reproduce the diffraction processes, as well as the sample and detector geometry, forward modelling is significantly better at determining the orientation in the presence of noisy diffraction patterns. The comparison metric between the predicted and experimental diffraction images (typically the dot product or sum of squared differences) exhibits a continuous degradation with increasing noise, as opposed to the catastrophic degradation exhibited when using inverse methods, as shown in Figure 11.3.

Despite the superior indexing of forward modelling methods, the solution of the inverse problem is still the dominant method. The real-time indexing of the inverse approach is desirable when noise is not an issue, whereas forward modelling approaches are regarded as post-processing techniques due to the heavy computational requirements. In the remainder of this section we will describe the contributions of Papers 6 and 7, which respectively treat the forward and inverse approaches.

11.2 Improvements to Forward Modelling Based Indexing

In forward modelling based pattern indexing, template patterns are generated for a set of orientations, and compared against the experimentally observed diffraction pattern. The best orientation is the one which has the best score according to a chosen comparison metric. Ideally, we would like to test every possible orientation against the experimental pattern. Clearly, since $SO(3)$ is a continuous space this is not possible, since it would require an infinite number of orientations. Instead, we must test a finite number by discretizing $SO(3)$. The pertinent question is then: how should we discretize, and how do we measure the quality of a discretization? This is the question we answer in Paper 6.

The fiducial representation of an orientation is as a unit quaternion [138, 25]. Briefly, a quaternion is a four dimensional vector of the form $\vec{q} = \{w, ix, jy, kz\}$, where w, x, y and z are real numbers, and i, j and k are imaginary numbers which generalize the better-known complex numbers. Quaternions with unit norm represent points on a four-dimensional (hyper)sphere. This space is formally known as $\mathbb{S}^3$, and is a double covering of $SO(3)$, that is, $-\vec{q} = \vec{q}$ (the negation of a quaternion represents the same orientation). This becomes evident with the formula to convert a quaternion to a rotation matrix:

$$\mathbf{U} = \begin{bmatrix} w^2 + x^2 - y^2 - z^2 & 2xy - 2wz & 2xz + 2wy \\ 2xy + 2wz & w^2 - x^2 + y^2 - z^2 & 2yz - 2wx \\ 2xz - 2wy & 2yz + 2wx & w^2 - x^2 - y^2 + z^2 \end{bmatrix} \tag{11.1}$$

Since every element of $\mathbf{U}$ is a linear sum of products of exactly two elements of $\vec{q}$, the sign of $\vec{q}$ has no importance. Thus, the creation of a good set of orientations is equivalent to creating a good set of points on $\mathbb{S}^3$ with antipodal symmetry (negation symmetry).

One of the principal advantages of using the quaternion representation is that it permits an exact calculation of the quality of a set of orientations using methods from computational geometry. We will first define a measure of quality. As stated, unit quaternions as simply points on a 4D sphere ($\mathbb{S}^3$). Given a set of points $\mathbf{P} = \{\vec{p_1}, \vec{p_2}, \ldots, \vec{p_N}\} \in \mathbb{S}^d$, the covering radius, $\theta_{\max}$, is defined as the largest angular distance between any point in $\mathbb{S}^d$ and $\mathbf{P}$, that is:

$$\theta_{\max} = \max_{x \in \mathbb{S}^d} \min_{\vec{p} \in \mathbf{P}} \arccos\left(\langle x, \vec{p} \rangle\right) \tag{11.2}$$

where $\langle x, \vec{p} \rangle$ denotes the inner product of x and $\vec{p}$. The covering radius is the best measure of quality, as it determines the maximum possible error between between $\mathbf{P}$ and $\mathbb{S}^d$. In practical terms, it means that we minimize the maximum orientational error of an experimental observation during pattern indexing.

The covering radius can be found using the Voronoi diagram of $\mathbf{P}$. In a spherical space, the Voronoi cell of point $\vec{p_i}$, denoted $\mathrm{Vor}(\vec{p_i})$, consists of all points of $\mathbb{S}^d$ that are at least as close to $\vec{p_i}$ as to any other $\vec{p_j}$. More specifically:

$$\mathrm{Vor}(\vec{p_i}) = \{x \in \mathbb{S}^d \mid \arccos\langle x, \vec{p_i} \rangle \leq \arccos\langle x, \vec{p_j} \rangle \ \ \forall j\} \tag{11.3}$$

Since the vertices of the Voronoi cells are the points which locally maximize the angular distance from **P**, the covering radius is determined by the Voronoi vertex that lies furthest from **P**. The spherically constrained Voronoi diagram is not easy to calcuate, but each Voronoi vertex lies at the circumcentre of a simplex in the Delaunay triangulation. This is simply the property of duality between the Voronoi diagram and the Delaunay triangulation. The spherically constrained Delaunay triangulation, though, is not easy to calculate directly either; Euclidean-space algorithms will permit triangulations across the interior of the hypersphere, whereas we want a triangulation of the surface. Instead, we exploit a feature of spherical geometry: the vertices of the spherically constrained Delaunay triangulation are the same as those of the convex hull. Whilst this calculation sounds long-winded, it is fairly simple:

1 Calculate the convex hull of **P**.
2 From Conv(**P**) we have DT(**P**), the Delaunay triangulation of **P**.
3 Calculate the circumcentre of each simplex in DT(**P**).
4 The maximum circumradius is the covering radius of **P**.

With the covering radius defined and a method to calculate it, we can compare different approaches for generating spherical point sets in a quantitative manner. Figure 11.4 shows a selection of point sets on the sphere. Spherical point sets are shown due to the difficulty of visualizing a hypersphere, but also form the basis of orientation set generation methods. The HEALPix [139] method is used to create infinite refinable orientation sequences [142], which has been used for X-ray diffraction pattern indexing [134]. Roşca et al. adapt the Lambert projection method [140] to create *isochoric* (equal-volume) orientation sets [143], which are widely used in pattern indexing [135, 136, 144].

A striking feature is how much smaller the optimal covering radius is compared to the other, more commonly used methods. Furthermore, we note that the set with the optimal Riesz energy, defined as:

$$E_s(\mathbf{P}) = \sum_{i \neq j}^{N} \frac{1}{|\vec{p_i} - \vec{p_j}|^s} \tag{11.4}$$

where s is a constant, strongly resembles the optimal covering, both visually and in terms of the covering radius. In Paper 6, we describe the generation of nearly-optimal orientation sets by minimizing the covering radius. The Riesz energy is used as a stepping stone, since it is a differentiable function of the points, which makes the problem significantly easier. At the time of writing, the resulting point sets have a smaller error term than any other in the literature.

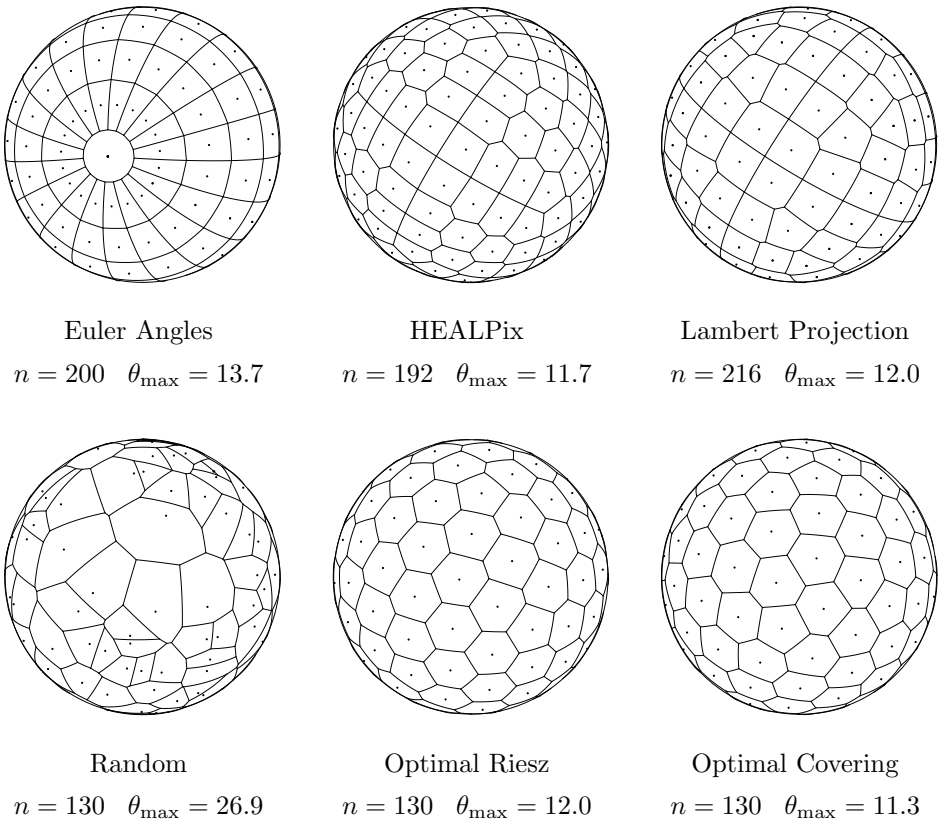


Figure 11.4. Different methods for distributing points on the sphere. The Voronoi cell of each point is also shown. The number of points is given by n, and $\theta_{\max}$ denotes the maximum simplex circumradius (the covering radius). It is well known that random points and linearly-stepped Euler angles result in a poor distribution. Optimal coverings, though, which achieve a significantly smaller error, are less well studied. HEALPix, Lambert, Riesz and covering data from references [139, 140, 141, 35].

11.3 Improvements to Inverse Problem Based Indexing

As stated previously, forward modelling based pattern indexing gives superior results, but at the cost of a significantly increased running time. When noise is not an issue, solution of the inverse problem is much faster. Here, we describe a small contribution to FELIX, an inverse pattern indexing method for X-ray Free Electron Laser (XFEL) diffraction data, presented in Paper 7.

FELIX builds upon the principles developed in GrainSpotter [145] (an indexing method for synchrotron diffraction), using the frustum ray-tracing approach developed by Kazantsev et al. [146]. Instead of testing every crystalline orientation against the experimental data, the orientation is determined by identifying the individual diffraction spots in the experimental data, and implicitly solving a system of nonlinear equations. For the full details of the method, we refer to the GrainSpotter [145] and FELIX [147] papers, but we will briefly describe the indexing procedure here.

Given a crystal structure, crystal orientation, beam wavelength, and the geometry of the experimental setup, we can calculate all scatter vectors, as well as the positions

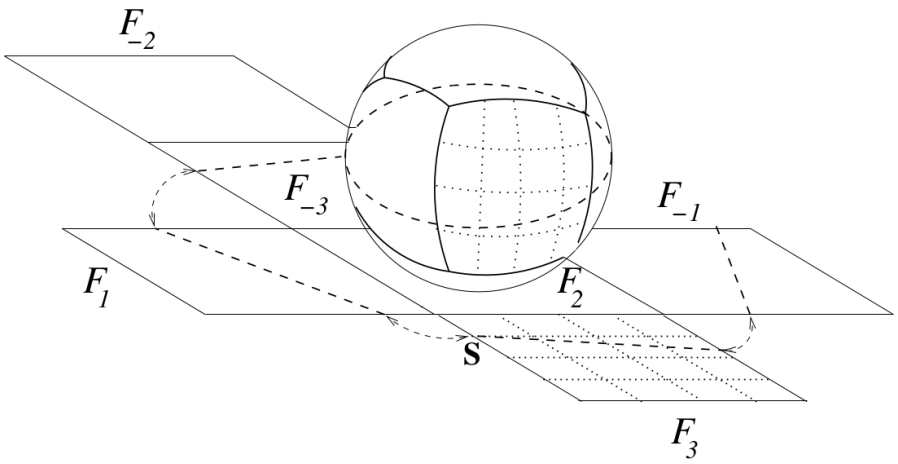


Figure 11.5. Gnomonic projection of a geodesic (great circle) on $\mathbb{S}^2$ (a 2D sphere). The sphere is bounded by the six faces of the cube (F_1, F_2, F_3 and their opposite faces, F_{-1}, F_{-2}, F_{-3}). Projection of the geodesic from the sphere surface produces a series of straight line segments on the cube faces. The square grid discretization, shown on face F_3, is used to identify regions where multiple geodesics intersect. Figure reproduced from paper by Kazantsev et al. [146].

on the detector of the associated diffraction spots. To solve the inverse problem, we must find the crystal orientation from the diffraction spots. A single diffraction spot, though, does not uniquely determine the orientation. For every diffraction spot there is a set of orientations which can produce it; it can be shown that this set is a geodesic in $\mathbb{S}^3$, i.e. a great circle in the space of unit quaternions. If we calculate the geodesics of every diffraction spot, the most likely crystal orientation lies in a region where multiple geodesics intersect. Determination of the regions of intersection proceeds by gnomonic projection of the geodesics.

Figure 11.5 shows a gnomonic projection of a geodesic on $\mathbb{S}^2$. The sphere $\mathbb{S}^2$ is bounded by a cube, whose six faces are folded out into a net in Figure 11.5. By gnomonic projection, the geodesics become straight lines on the cube faces. The cube faces are discretized using a square grid, each of which has a counter variable. For each grid square which a projected geodesic intersects, the counter is incremented. The grid squares with high counter values are potential candidates for the crystal orientation.

The use of $\mathbb{S}^2$ in Figure 11.5 is merely for illustrative purposes. In $\mathbb{S}^3$, the cube faces correspond to frusta, which are the eight cubic 'faces' of a hypercube. In order to increment the counters of the discretized frusta, we need to determine which frusta are visited, as well as the entry and exit points in each of these. Fast determination entry-points and exit-points is the contribution of this thesis to inverse problem based indexing.

Let S be a geodesic on $\mathbb{S}^3$ parametrised by two orthogonal quaternions, P and Q:

$$S = P\cos t + Q\sin t \tag{11.5}$$

Kazantsev et al. [146] use a convex hull to determine which frusta are visited, as shown

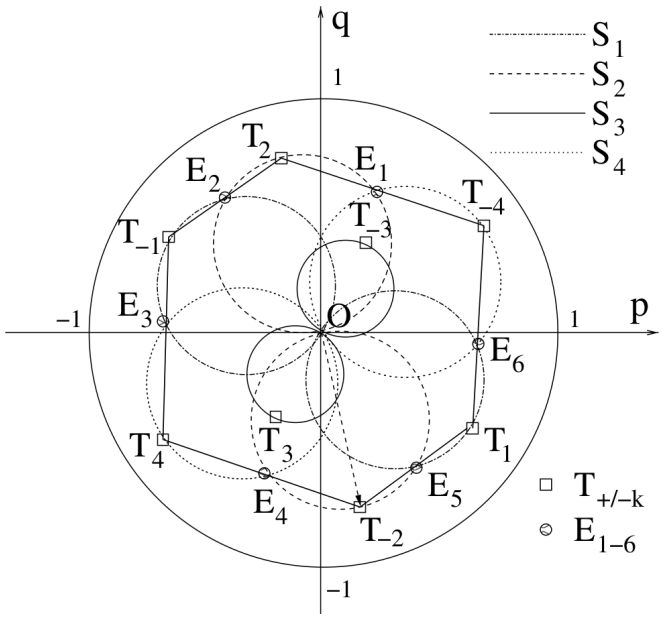


Figure 11.6. Convex hull method for determination of visited frusta. Each vertex T_k has coordinates (P_k, Q_k). A frustum is visited if the corresponding vertex lies on the convex hull. The entry-exit points, E_{1-6}, lie at the intersection of the circles with the origin and T_k at antipodally opposite points. Figure reproduced from paper by Kazantsev et al. [146].

in Figure 11.6. This is a clever use of the convex hull method, which scales well with an increasing number of dimensions. When restricted to the four dimensional space of quaternions, a simpler procedure gives better performances.

Given that S crosses from frustum i to frustum j, the i^{th} and j^{th} components of S must be equal in magnitude:

$$|S_i| = |S_j| \tag{11.6}$$

If the frusta i and j are visited, then the i^{th} and j^{th} components of S must also be the largest components at crossing point:

$$|S_i| \geq |S_k| \;\; \forall k \tag{11.7}$$

We can then write:

$$P_i \cos t + Q_i \sin t = P_j \cos t + Q_j \sin t \tag{11.8}$$

By rearrangement, we obtain:

$$\sin t = \frac{1}{c}(P_i - P_j) \tag{11.9}$$

$$\cos t = \frac{1}{c}(Q_j - Q_i) \tag{11.10}$$

where t is the angle where S crosses from frustum i to frustum j, and $c = \sqrt{(P_i - P_j)^2 + (Q_j - Q_i)^2}$ is a scaling factor. On $\mathbb{S}^3$, there are six unique possible frustum crossings to investigate:

i	1	1	1	2	2	3
j	2	3	4	3	4	4

Any combination of i and j which does not fulfil the condition in Equation (11.7) is not a frustum crossing. Next, let $\sin' t = P_i - P_j$ and $\cos' t = Q_j - Q_i$ (i.e. letting $c = 1$), and feed these values into Equation (11.5); the condition in Equation (11.7) can still be checked without the correct value of c, since only the *relative* magnitudes matter. Gnomonic projection is also scale independent (by definition). As a consequence, we can determine the visited frusta and their crossing points on S using *only multiplication, subtraction and addition operations*. No trigonometric (sine, cosine) or square root operations are needed. In a test of the CPU implementation, this method is an order of magnitude faster than the convex hull method [148]. More importantly, the method can be implemented with very little branching (`IF`-statements), which results in significant speedups on a GPU implementation.

References

[1] Kraitchik M 1942 *Mathematical Recreations* (W. W. Norton, New York)
[2] Gusfield D and Irving R W 1989 *The stable marriage problem: structure and algorithms* (MIT press)
[3] Segev D L, Gentry S E, Warren D S, Reeb B and Montgomery R A 2005 *JAMA* **293** 1883–1890
[4] Kuhn H W 1955 *Naval Research Logistics Quarterly* **2** 83–97
[5] Edmonds J 1965 *J. Res. Nat. Bur. Standards* **B69** 125–130
[6] McKay B D and Piperno A 2014 *J. Sym. Comp.* **60** 94–112
[7] Weinberg L 1966 *IEEE Trans. Circuit Theory* **13** 142–148
[8] Falk M L and Langer J S 1998 *Phys. Rev. E* **57**(6) 7192–7205
[9] Roger H and Charles R J 1994 Topics in matrix analysis
[10] Autonne L 1902 *Bull. Soc. Math. France* **30** 121–134
[11] Wintner A and Murnaghan F D 1931 *Proc. Natl. Acad. Sci.* **17** 676–678
[12] Golub G and Kahan W 1965 *SIAM J. Soc. Industr. Appl. Math.* **B2** 205–224
[13] Higham N J 1986 *SIAM J. Sci. Stat. Comp.* **7** 1160–1174
[14] Higham N J and Noferini V 2016 *Numerical Algorithms* **73** 349–369
[15] Karney C F 2007 *J. Mol. Graph. Model.* **25** 595 – 604 ISSN 1093-3263
[16] Wahba G 1965 *SIAM Rev.* **7** 409–409
[17] Keat J 1977 Analysis of least-squares attitude determination routine DOAOP Tech. rep. Technical Report CSC/TM-77/6034, Comp. Sc. Corp
[18] Lerner G M 1978 Three-axis attitude determination *Spacecraft Attitude Determination and Control* ed JR W (Dordrecht: Kluwer Academic) chap 12, pp 420–428
[19] Faugeras O D and Hebert M 1983 A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 2* IJCAI'83 pp 996–1002
[20] Horn B K P 1987 *J. Opt. Soc. Am. A* **4** 629–642
[21] Diamond R 1988 *Acta Cryst.* **A44** 211–216
[22] Kearsley S K 1989 *Acta Cryst.* **A45** 208–210
[23] Liu P, Agrafiotis D K and Theobald D L 2010 *J. Comp. Chem.* **31** 1561–1563 ISSN 1096-987X
[24] Kabsch W 1978 *Acta Cryst.* **A34** 827–828
[25] Altmann S L 2005 *Rotations, Quaternions, and Double Groups* (Courier Dover Publications)
[26] Theobald D L 2005 *Acta Cryst.* **A61** 478–480
[27] Diamond R 1990 *Acta Cryst.* **A46** 423–423
[28] Higham N J and Noferini V 2016 Polar decomposition of a real 3x3 matrix - Source Code Repository http://github.com/higham/polar-decomp-3by3
[29] de Berg M, Cheong O, van Kreveld M and Overmars M 2008 *Computational Geometry: Algorithms and Applications* (Springer) ISBN 978-3-540-77974-2
[30] Devadoss S L and O'Rourke J 2011 *Discrete and computational geometry* (Princeton University Press)
[31] Delaunay B 1934 *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* **7** 1–2
[32] Voronoï G 1908 *Journal für die reine und angewandte Mathematik* **134** 198–287
[33] Grünbaum B, Klee V, Perles M A and Shephard G C 1967 *Convex polytopes* vol 16 (Springer)
[34] Rockafellar R T 1970 *Convex Analysis* (Princeton University Press)
[35] Hardin R H, Sloane N J A and Smith W D - Spherical codes, book in preparation http://neilsloane.com/coverings/
[36] Stukowski A 2009 *Modelling Simul. Mater. Sci. Eng.* **18** 015012
[37] Kelchner C L, Plimpton S J and Hamilton J C 1998 *Phys. Rev. B* **58** 11085–11088
[38] Honeycutt J D and Andersen H C 1987 *J. Phys. Chem.* **91** 4950–4963
[39] Stukowski A 2012 *Modelling Simul. Mater. Sci. Eng.* **20** 045021
[40] Larsen P M, Schmidt S and Schiøtz J 2016 *Modelling Simul. Mater. Sci. Eng.* **24** 055007

[41] Möller J J and Bitzek E 2016 *MethodsX* **3** 279–288
[42] Buehler M 2008 *Atomistic modeling of materials failure* (Springer Science & Business Media)
[43] Plimpton S 1995 *J. Comp. Phys.* **117** 1–19
[44] Bulatov V and Cai W 2006 *Computer simulations of dislocations* (Oxford University Press)
[45] Li J 2003 Central symmetry parameter http://li.mit.edu/A/Graphics/A/Doc/CentralSymmetry.pdf
[46] Li J 2003 *Modelling Simul. Mater. Sci. Eng.* **11** 173
[47] Kolmogorov V 2009 *Math. Program. Comp.* **1** 43–67
[48] Dezső B, Jüttner A and Kovács P 2011 *Elec. Notes Theor. Comp. Sci.* **264** 23–45
[49] Karlsruhe FIZ Inorganic Crystal Structure Database URL http://icsd.fiz-karlsruhe.de
[50] Jain A, Ong S P, Hautier G, Chen W, Richards W D, Dacek S, Cholia S, Gunter D, Skinner D, Ceder G and Persson K A 2013 *APL Materials* **1** 011002 ISSN 2166532X
[51] Nomad repository URL http://repository.nomad-coe.eu/
[52] AFLOWLIB - Automatic FLOW for Materials Discovery URL http://aflowlib.org/
[53] Saal J E, Kirklin S, Aykol M, Meredig B and Wolverton C 2013 *JOM* **65** 1501–1509
[54] Lyakhov A O, Oganov A R, Stokes H T and Zhu Q 2013 *Comp. Phys. Comm.* **184** 1172–1182
[55] Valle M and Oganov A R 2010 *Acta Cryst.* **A66** 507–517
[56] Bartók A P, Kondor R and Csányi G 2013 *Phys. Rev. B* **87** 184115
[57] Sadeghi A, Ghasemi S A, Schaefer B, Mohr S, Lill M A and Goedecker S 2013 *J. Chem. Phys.* **139** 184118
[58] Ferré G, Maillet J B and Stoltz G 2015 *J. Chem. Phys.* **143** 104114
[59] Hundt R, Schön J and Jansen M 2006 *J. Appl. Cryst.* **39** 6–16
[60] Lonie D C and Zurek E 2012 *Comp. Phys. Comm.* **183** 690–697
[61] Zhu L, Amsler M, Fuhrer T, Schaefer B, Faraji S, Rostami S, Ghasemi S A, Sadeghi A, Grauzinyte M, Wolverton C and Goedecker S 2016 *The Journal of Chemical Physics* **144** 034203
[62] Rupp M, Tkatchenko A, Müller K R and Von Lilienfeld O A 2012 *Physical review letters* **108** 058301
[63] Hansen K, Biegler F, Ramakrishnan R, Pronobis W, Von Lilienfeld O A, Müller K R and Tkatchenko A 2015 *J. Phys. Chem. Lett* **6** 2326–2331
[64] Mills-Tettey G A, Stentz A and Dias M B 2007 The dynamic hungarian algorithm for the assignment problem with changing costs Tech. rep. Carnegie Mellon University
[65] Climer S and Zhang W 2006 *Journal of Machine Learning Research* **7** 919–943
[66] Novoselov K S, Geim A K, Morozov S V, Jiang D, Zhang Y, Dubonos S V, Grigorieva I V and Firsov A A 2004 *Science* **306** 666–669
[67] Steinhardt P J, Nelson D R and Ronchetti M 1983 *Phys. Rev. B* **28** 784
[68] Lazar E A, Han J and Srolovitz D J 2015 *Proc. Natl. Acad. Sci. USA* **112** E5769–E5776
[69] Steinitz E 1916 *Polyeder und Raumeinteilungen* (Teubner)
[70] Huddleston B 2017 Single defect velocity in ovitos - teaching materials URL http://icme.hpc.msstate.edu/mediawiki/index.php/Single_Defect_Velocity_in_Ovitos
[71] Zhong L, Sansoz F, He Y, Wang C, Zhang Z and Mao S X 2017 *Nature Mater.* **16** 439–445
[72] Yamanaka A, McReynolds K and Voorhees P W 2017 *Acta Mater.* **133** 160–171
[73] Eder S J, Cihak-Bayr U, Bianchi D, Feldbauer G and Betz G 2017 *ACS Appl. Mater. & Interfaces* **9** 13713–13725
[74] Huang P Y, Ruiz-Vargas C S, van der Zande A M, Whitney W S, Levendorf M P, Kevek J W, Garg S, Alden J S, Hustedt C J, Zhu Y, Park J, McEuen P L and Muller D A 2011 *Nature* **469** 389–392 ISSN 0028-0836
[75] Banhart F, Kotakoski J and Krasheninnikov A V 2010 *ACS Nano* **5** 26–41
[76] Rani P and Jindal V 2013 *RSC Advances* **3** 802–812
[77] Wang X, Sun G, Routh P, Kim D H, Huang W and Chen P 2014 *Chem. Soc. Rev.* **43** 7067–7098
[78] Ophus C, Shekhawat A, Rasool H and Zettl A 2015 *Phys. Rev. B* **92** 205402
[79] Stuart S J, Tutein A B and Harrison J A 2000 *J. Chem. Phys.* **112** 6472–6486

[80] Galindo P L, Kret S, Sanchez A M, Laval J Y, Yez A, Pizarro J, Guerrero E, Ben T and Molina S I 2007 *Ultramicroscopy* **107** 1186 – 1193 ISSN 0304-3991
[81] Kronberg M and Wilson F 1949 *Trans. Metall. AIME* **185** 501–514
[82] Brandon D, Ralph B, Ranganathan S t and Wald M 1964 *Acta Metall.* **12** 813–821
[83] Lin P, Palumbo G, Erb U and Aust K 1995 *Scripta Metall. Mater.* **33** 1387–1392
[84] Randle V 2001 *Materials Characterization* **47** 411–416
[85] Randle V 2010 *Mater. Sci. Tech.* **26** 253–261
[86] Randle V 2010 *Mater. Sci. Tech.* **26** 774–780
[87] Stradi D, Jelver L, Smidstrup S and Stokbro K 2017 *J. Phys. Cond. Mater.* **29** 185901
[88] LeGoues F, Copel M and Tromp R 1990 *Phys. Rev. B* **42** 11690
[89] Tersoff J 1993 *Appl. Phys. Lett.* **62** 693–695
[90] Ihli J, Clark J N, Côté A S, Kim Y Y, Schenk A S, Kulak A N, Comyn T P, Chammas O, Harder R J, Duffy D M *et al.* 2016 *Nature Comm.* **7** 11878
[91] Albrecht S M, Higginbotham A, Madsen M, Kuemmeth F, Jespersen T S, Nygård J, Krogstrup P and Marcus C 2016 *Nature* **531** 206–209
[92] Takei S, Fregoso B M, Hui H Y, Lobos A M and Sarma S D 2013 *Phys. Rev. Lett.* **110** 186803
[93] Krogstrup P, Ziino N, Chang W, Albrecht S, Madsen M, Johnson E, Nygard J, Marcus C, Jespersen T *et al.* 2015 *Nature Mater.* **14** 400–406
[94] Hart G L, Blum V, Walorski M J and Zunger A 2005 *Nat. Mater.* **4** 391–394
[95] Sanchez J, Ducastelle F and Gratias D 1984 *Physica A* **128** 334 – 350 ISSN 0378-4371
[96] Wallace G K 1992 *IEEE Trans. on Consum. Electron.* **38** xviii–xxxiv
[97] Wiegand T, Sullivan G J, Bjontegaard G and Luthra A 2003 *IEEE Trans. Cir. and Sys. for Video Technol.* **13** 560–576
[98] Nelson L J, Hart G L W, Zhou F and Ozoliņš V 2013 *Phys. Rev. B* **87**(3) 035125
[99] Tan T L, Wang L L, Johnson D D and Bai K 2012 *Nano Letters* **12** 4875–4880
[100] Kozlov S M, Kovács G, Ferrando R and Neyman K M 2015 *Chemical Science* **6** 3868–3880
[101] Seko A, Koyama Y and Tanaka I 2009 *Phys. Rev. B* **80**(16) 165122
[102] Lerch D, Wieckhorst O, Hart G L, Forcade R W and Müller S 2009 *Modelling Simul. Mater. Sci. Eng.* **17** 055003
[103] van de Walle A and Ceder G 2002 *J. Phase Equilib.* **23** 348
[104] Van De Walle A, Asta M and Ceder G 2002 *Calphad* **26** 539–553
[105] Zarkevich N A, Tan T L and Johnson D D 2007 *Phys. Rev. B* **75** 104203
[106] Zarkevich N A, Tan T L, Wang L L and Johnson D D 2008 *Phys. Rev. B* **77** 144208
[107] Teeriniemi J, Melander M, Lipasti S, Hatz R and Laasonen K 2017 *J. Phys. Chem. C* **121** 1667–1674
[108] Tikhonov A N, Arsenin V I A and John F 1977 *Solutions of ill-posed problems* vol 14 (Winston Washington, DC)
[109] Zarkevich N A and Johnson D D 2004 *Phys. Rev. Lett.* **92**(25) 255702
[110] Nelson L J, Ozoliņš V, Reese C S, Zhou F and Hart G L W 2013 *Phys. Rev. B* **88**(15) 155105
[111] Candes E J, Romberg J K and Tao T 2006 *Comm. Pure Appl. Math.* **59** 1207–1223
[112] Sai Gautam G, Canepa P, Abdellahi A, Urban A, Malik R and Ceder G 2015 *Chemistry of Materials* **27** 3733–3742
[113] Bertsimas D, King A and Mazumder R 2016 *Ann. Stat.* **44** 813–852
[114] Natarajan B K 1995 *SIAM J. Comp.* **24** 227–234
[115] Chen R, Maawad E, Knapp M, Ren S, Beran P, Witter R and Hempelmann R 2016 *RSC Advances* **6** 65112–65118
[116] Cambaz M A, Vinayan B, Clemens O, Munnangi A R, Chakravadhanula V S K, Kubel C and Fichtner M 2016 *Inorganic Chem.* **55** 3789–3796
[117] Ren S, Chen R, Maawad E, Dolotko O, Guda A A, Shapovalov V, Wang D, Hahn H and Fichtner M 2015 *Advanced Science* **2**
[118] Gurobi Optimization I 2017 Gurobi optimizer reference manual URL `http://www.gurobi.com`

[119] Huang W, Kitchaev D A, Dacek S T, Rong Z, Urban A, Cao S, Luo C and Ceder G 2016 *Phys. Rev. B* **94**(13) 134424
[120] Boros E and Hammer P L 2002 *Discrete Applied Mathematics* **123** 155 – 225 ISSN 0166-218X
[121] Daw M S and Baskes M I 1984 *Phys. Rev. B* **29** 6443
[122] Jacobsen K W, Stoltze P and Nørskov J K 1996 *Surf. Sci.* **366** 394–402
[123] Adamo C and Barone V 1998 *J. Chem. Phys.* **108** 664–675
[124] Lee C, Yang W and Parr R G 1988 *Phys. Rev. B* **37** 785
[125] Kogure T 2002 *American Mineralogist* **87** 1678–1685
[126] Abe E, Yan Y and Pennycook S J 2004 *Nature Mater.* **3** 759–767
[127] Tanaka M, Saito R, Ueno K and Harada Y 1980 *J. Electron. Microsc.* **29** 408–412
[128] Schotte F 2000 *Structural Dynamics of Photosensitive Proteins* Ph.D. thesis Heinrich-Heine-Universitt Dsseldorf
[129] Rauch E F, Portillo J, Nicolopoulos S, Bultreys D, Rouvimov S and Moeck P 2010 *Zeitschrift für Kristallographie International journal for structural, physical, and chemical aspects of crystalline materials* **225** 103–109
[130] Hough P V 1959 Machine analysis of bubble chamber pictures *International conference on high energy accelerators and instrumentation* vol 73 p 2
[131] Duda R O and Hart P E 1972 *Comm. ACM* **15** 11–15
[132] Wright S I, Zhao J W and Adams B L 1991 *Texture, Stress, and Microstructure* **13** 123–131
[133] Rauch E and Dupuy L 2005 *Archives of Metallurgy and Materials* **50** 87–99
[134] Li S F and Suter R M 2013 *J. Appl. Cryst.* **46** 512–524
[135] Chen Y H, Park S U, Wei D, Newstadt G, Jackson M A, Simmons J P, De Graef M and Hero A O 2015 *Microsc. Microanal.* **21** 739–752
[136] Singh S and De Graef M 2017 *Microsc. Microanal.* **23** 1–10
[137] Wright S I, Nowell M M, Lindeman S P, Camus P P, De Graef M and Jackson M A 2015 *Ultramicroscopy* **159** 81–94
[138] Hamilton W R 1844 *Phil. Mag.* **25** 10–13
[139] Gorski K M, Hivon E, Banday A, Wandelt B D, Hansen F K, Reinecke M and Bartelmann M 2005 *The Astrophysical Journal* **622** 759
[140] Roşca D and Plonka G 2011 *J. Comp. Appl. Math.* **236** 1033–1041
[141] Wales D J and Ulker S 2006 *Phys. Rev. B* **74**(21) 212101
[142] Yershova A, Jain S, LaValle S M and Mitchell J C 2010 *Int. J. Robot. Res.* **29** 801–812
[143] Roşca D, Morawiec A and De Graef M 2014 *Modelling Simul. Mater. Sci. Eng.* **22**
[144] Ram F, Wright S, Singh S and De Graef M 2017 *Ultramicroscopy* **181** 17–26
[145] Schmidt S 2014 *J. Appl. Cryst.* **47** 276–284
[146] Kazantsev I G, Schmidt S and Poulsen H F 2009 *Inverse Problems* **25** 105009
[147] Beyerlein K R, White T A, Yefanov O, Gati C, Kazantsev I G, Nielsen N F G, Larsen P M, Chapman H N and Schmidt S 2017 *J. Appl. Cryst.* **50**
[148] Nielsen N F G 2016 Personal communication

Paper 1

Robust structural identification via polyhedral template matching
Larsen P M, Schmidt S and Schiøtz J
Modelling and Simulation in Materials Science and Engineering
Volume 24, Number 5, Pages 055007

IOP Publishing Modelling and Simulation in Materials Science and Engineering

Modelling Simul. Mater. Sci. Eng. **24** (2016) 055007 (18pp) doi:10.1088/0965-0393/24/5/055007

Robust structural identification via polyhedral template matching

Peter Mahler Larsen, Søren Schmidt and Jakob Schiøtz

Department of Physics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

E-mail: schiotz@fysik.dtu.dk

Received 18 February 2016, revised 19 April 2016
Accepted for publication 25 April 2016
Published 13 May 2016

Abstract

Successful scientific applications of large-scale molecular dynamics often rely on automated methods for identifying the local crystalline structure of condensed phases. Many existing methods for structural identification, such as common neighbour analysis, rely on interatomic distances (or thresholds thereof) to classify atomic structure. As a consequence they are sensitive to strain and thermal displacements, and preprocessing such as quenching or temporal averaging of the atomic positions is necessary to provide reliable identifications. We propose a new method, polyhedral template matching (PTM), which classifies structures according to the topology of the local atomic environment, without any ambiguity in the classification, and with greater reliability than e.g. common neighbour analysis in the presence of thermal fluctuations. We demonstrate that the method can reliably be used to identify structures even in simulations near the melting point, and that it can identify the most common ordered alloy structures as well. In addition, the method makes it easy to identify the local lattice orientation in polycrystalline samples, and to calculate the local strain tensor. An implementation is made available under a Free and Open Source Software license.

Keywords: structure classification, atomic-scale simulations, crystal structure, atomic systems visualization, analysis of molecular dynamics, neighbor analysis, alloy classification

(Some figures may appear in colour only in the online journal)

1. Introduction

Often, the most challenging part of a molecular dynamics (MD) simulation is analysing the large amounts of data generated. For simulations within condensed matter physics,

algorithms for automatic determination of local structure are often necessary, for example to identify crystalline phases, defects in the crystal structure, or structural motifs in non-crystalline samples. One method is to use the energy of the individual atoms and identify defect atoms by a threshold value of the energy. Another possibility is the centro-symmetry parameter, which is zero in many crystalline phases, but takes a non-zero value at defects such as dislocation cores [1]. Both methods are reasonably effective at identifying atoms near defects if the temperature is not too high, but do not provide additional information about the local structure.

A very popular analysis method is common neighbour analysis (CNA) [2, 3], which classifies bonds between atoms according the local bonding structure, and uses this to assign a local crystalline structure to the atoms. The CNA has successfully been used to identify dislocations and grain boundaries in deforming polycrystalline metals [4–6], local ordering in amorphous phases [7, 8], and the competition between crystalline and icosahedral order in nanoparticles [9, 10].

The CNA method analyses the bonds between common neighbours of two bonded atoms. It relies heavily on the concept of two atoms being 'bonded' or 'neighbours', and thus needs a strict definition of this; typically in the form of a cut-off distance defining if two atoms are neighbours. Such a cut-off distance is, by necessity, somewhat arbitrary. Furthermore, thermal vibrations, coexistence of various phases, and fluctuations in the local density may all influence the result and either introduce noise in the analysis, or make it impossible to find a cutoff parameter useful for the entire system. For these reasons, Stukowski introduced the adaptive common neighbour analysis (ACNA) [11], where the cutoff distance is picked automatically, and individually, for each atom. While this makes the method significantly more robust, the ACNA still suffers from noise introduced by thermal vibrations, causing bonds to be sporadically broken or formed.

Recently, Lazar *et al* have introduced a method classifying the local structure by the topology of the Voronoi cell surrounding the atom [12]. While this makes the method more robust to thermal vibrations than the (ACNA), the method has difficulty distinguishing between the face-centered cubic (fcc) and hexagonal closed packed (hcp) structures.

In this paper, we introduce the polyhedral template matching (PTM), an approach which is similar in spirit to the method of Lazar *et al*. The gist of the method is that the convex hull formed by the set of neighbouring atoms describes the local structure around an atom. The convex hull is represented as a planar graph, and this graph is then used to classify the structure. As this method looks at a fixed number of neighbouring atoms around the atom being analysed, and as it does not employ a concept of bonds between these atoms, it is less sensitive to thermal fluctuations. In addition, the method assigns a well-defined order to the neighbours of an atom, making it much easier to define a local orientation or a local elastic strain without referring to an initial unstrained structure. An implementation of the method is available online [13].

The structure of the paper is as follows: in section 2, we review how structures can easily be evaluated against a template structure, given that mapping between the neighbouring atoms and the template atoms exists. In section 3 we describe two methods for selecting the correct neighbouring atoms. In section 4 we show how the convex hull of the neighbouring atoms can be used to define a low number of candidate templates and mappings, and present the resulting algorithm for structure identification. In section 5, we benchmark the algorithm. In section 6 we extend it to ordered alloys. Finally, in sections 7 and 8, we illustrate how the method can provide information about local lattice orientation and local elastic strain, the former at no additional computational cost.

2. Template matching

In this section we will describe how template matching can be used to choose the best structural match, given that a point-to-point correspondence between simulated and template structures exists. The process of finding the point-to-point correspondences is described in the next section. A commonly used measure of similarity between two point sets is the root-mean-square deviation (RMSD). Given two sets of points $\mathbf{v}$ and $\mathbf{w}$, the RMSD is defined as:

$$\mathrm{RMSD}(\mathbf{v},\mathbf{w}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\vec{v}_i - \vec{w}_i\|^2} \tag{1}$$

The superposition problem is that of finding a translation and a rotation of $\mathbf{w}$ and a scaling of $\mathbf{v}$ which minimizes the RMSD. This is equivalent to Horn's scale-asymmetric formulation [14]. It can be shown that the optimal translation is equivalent to bringing the barycentre of each point set to the origin. The optimal rotation and scaling are given by:

$$\mathrm{RMSD}(\mathbf{v},\mathbf{w}) = \min_{s,\mathbf{Q}} \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|s[\vec{v}_i - \overline{\mathbf{v}}] - (\mathbf{Q}[\vec{w}_i - \overline{\mathbf{w}}]^T)^T\|^2} \tag{2}$$

where $\mathbf{Q}$ is a right-handed orthogonal matrix, $\overline{\mathbf{v}} = \frac{1}{N}\sum_{i=1}^{N}\vec{v}_i$ and $\overline{\mathbf{w}} = \frac{1}{N}\sum_{i=1}^{N}\vec{w}_i$ are the barycentres of $\mathbf{v}$ and $\mathbf{w}$ respectively, and s is the optimal scaling of $\mathbf{v}$. Finding $\mathbf{Q}$ is a well studied problem with many different solution methods; Theobald [15] provides a good exposition of the problem. Horn [14] describes a solution for finding s and shows that $\mathbf{Q}$ is independent of s. We can make the RMSD scale invariant by scaling $\mathbf{w}$ such that the mean distance of each point from the origin (after translation) is 1:

$$\mathrm{RMSD}(\mathbf{v},\mathbf{w}) = \min_{s,\mathbf{Q}} \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|s[\vec{v}_i - \overline{\mathbf{v}}] - \frac{1}{l(\mathbf{w})}(\mathbf{Q}[\vec{w}_i - \overline{\mathbf{w}}]^T)^T\right\|^2} \tag{3}$$

where:

$$l(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}\|\vec{w}_i - \overline{\mathbf{w}}\| \tag{4}$$

Suppose now that we want to determine the structure of a central atom and its neighbours. We wish to determine if it has simple cubic (SC), face-centred cubic (FCC), hexagonal close packed (HCP), icosahedral (ICO) or body-centred cubic (BCC) structure. For SC, FCC, HCP and ICO the positions of the first shell of neighbouring atoms are sufficient to identify the structure. For BCC the first two shells are required. Table 1 shows the number of atoms required for each structure. Correct identifying the shell to which a neighbouring atom belongs is nontrivial at high temperatures; this is discussed in section 3. Given a set of reference templates corresponding to the atom positions of the aforementioned structures, the template which best matches an atom and its neighbours is the template which minimizes the RMSD after superposition. In the above formulation, $\mathbf{v}$ contains the positions of the central atom and its neighbours and $\mathbf{w}$ contains those of the template. A scale-invariant RMSD serves two purposes; it avoids preferential weighting of smaller templates and avoids the need for selecting bond lengths.

As we have demonstrated here, the task of structural identification would be simple if the optimal point-to-point correspondences were known. Clearly, a brute force approach of testing all possible permutations of the neighbours is computationally infeasible. Fast determination

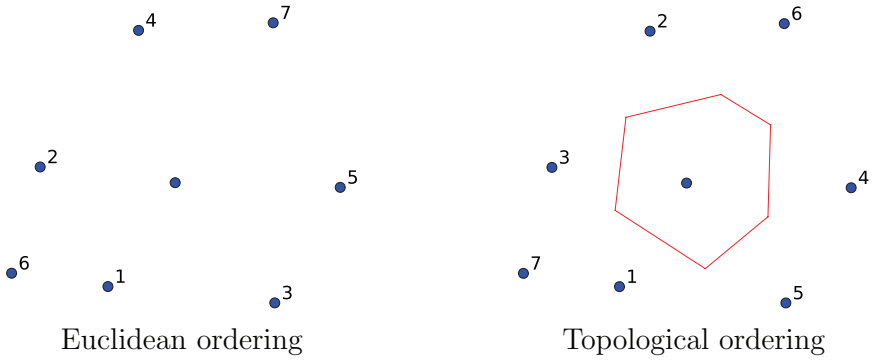


Figure 1. Neighbour atoms in a 2D hexagonal lattice ordered by Euclidean distance from the central atom (left) and by lexicographic ordering of the area of the polygonal interface between Voronoi cells and Euclidean distance (right). A 2D example is used here for ease of illustration. In $\mathbb{R}^2$, the 'polygonal' interfaces between Voronoi cells are line segments, and the interfacial area of each is simply the length.

of point-to-point correspondences is the main contribution of this work; the algorithm for doing this is developed in section 4.

3. Neighbouring atom shell identification

In order to use the template matching approach, we must correctly identify the nearest neighbours of each atom. In a perfect lattice, all atoms in the same neighbour shell lie at the same distance from a central atom, by definition. At low temperatures, thermal displacements are small enough such that the Euclidean distances from the central atom are sufficient to identify the shell numbers of neighbouring atoms. At high temperatures, the thermal displacements are large enough that an atom in the second neighbour shell can be closer to the central atom than one in the first neighbour shell. In this case, rather than using distance to order the neighbouring atoms, we can order them using the areas of the bounding polygons of the Voronoi cell of the central atom. Given a discrete collection of points $\mathbf{P} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_N\} \in \mathbb{R}^3$, the Voronoi cell of a point p_i consists of all points in $\mathbb{R}^3$ which are at least as close to p_i as to any other p_j. The boundary of a Voronoi cell can be defined by a set of polygons, each of which defines the interface to an adjacent Voronoi cell. Since we also wish to order neighbouring atoms whose Voronoi cells are not adjacent to that of the central atom, we use a lexicographical ordering of the interfacial areas *and* the distances:

$$\left(A_i, \frac{1}{d_i}\right) \geqslant \left(A_j, \frac{1}{d_j}\right) \quad \forall\, i < j \tag{5}$$

where A_i is the area of the polygonal interface between the Voronoi cells of the central atom and a neighbouring atom, and d_i is the Euclidean distance between them. In the case where the Voronoi cell of a neighbouring atom is not adjacent to that of the central atom, we assign an area of $A_i = 0$. We denote this ordering as the 'topological' ordering, which is demonstrated for a 2D example in figure 1.

4. Convex hulls

We have described how template matching can be used to find the best structural match, given a known point-to-point correspondence. In this section we will describe how convex hull

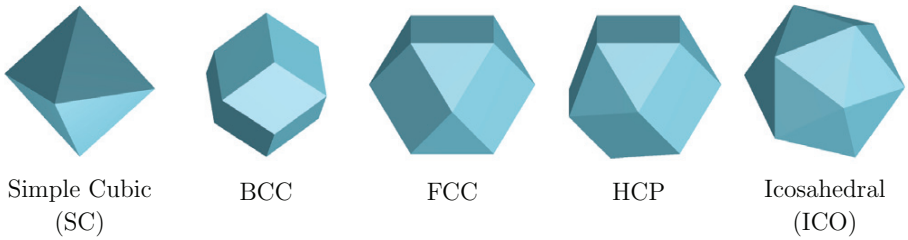


Figure 2. Convex hulls of the nearest neighbours of five different structures. For the BCC structure, the first two shells of nearest neighbours are included.

Table 1. Number of neighbouring atoms required to identify structures.

Structure	Neighbours required
SC	6
FCC	12
HCP	12
ICO	12
BCC	14

graphs can be used find all possible point-to-point correspondences in an efficient manner. A convex hull of a set of vertices is the smallest convex set which contains all the vertices. In $\mathbb{R}^3$ it can be described by a set of bounding planes. Figure 2 shows the convex hulls of the five different structural types we wish to match. The convex hull is the polyhedron formed by the nearest neighbours of a central atom.

Identification of the SC, FCC, HCP and ICO structures requires only the vertices of the first neighbour shell, which form a convex set. Two neighbour shells are required for identification of the BCC structure but, remarkably, these vertices also form a convex set. A stronger requirement of a convex hull is that it must consist only of simplicial facets, which in $\mathbb{R}^3$ are triangles; this is a *triangulation* of the surface of the convex hull. We can furthermore require that the triangulations fulfil the empty-sphere condition, that is, no vertex is contained inside the circumcircle of another simplicial facet. This is known as a *Delaunay* triangulation. Multiple Delaunay triangulations can exist for convex hulls with more than three coplanar points. Let us first consider the SC and ICO cases. The convex hulls of these structures consist only of triangular facets and the convex hull is therefore a unique triangulation. The FCC and HCP convex hulls consist of both equilateral triangular facets and perfect square facets. A perfect square has two equally valid triangulations which means the convex hull triangulation is not unique. Furthermore, the small atomic displacements can change the triangulation of the square facets. Figure 3 shows four different, equally valid, triangulations of the FCC convex hull.

The BCC convex hull consists only of rhombus facets. Whilst an unperturbed rhombus facet has a unique Delaunay triangulation, small changes in vertex positions can change the triangulation. We will consider both triangulations of each rhombus facet, since this will make our algorithm robust against relatively large atomic displacements.

We can represent the convex hull triangulations in graph form; vertices and edges in the convex hull triangulation correspond to vertices and edges in the graph. By representing a convex hull as a graph, we move from a metric space to a purely topological space. Steinitz's theorem [16] states that the skeleton of a three-dimensional convex polyhedron is a tri-connected planar graph. Conversely, any tri-connected planar graph has two embeddings in $\mathbb{R}^3$, i.e. it corresponds to two polyhedra that are mirror images of each other [17]. Figure 4 shows the

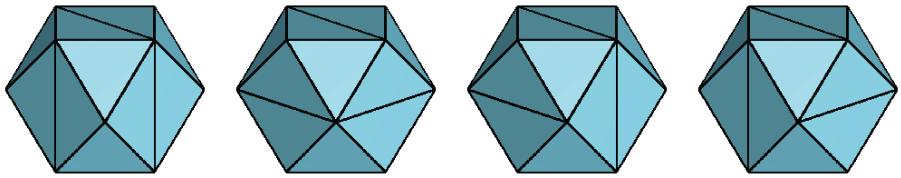


Figure 3. Four different triangulations of the FCC convex hull. The convex hull contains six square facets, which gives a total of 2^6 different triangulations.

planar graph representations of the convex hulls shown in figure 2. Since the FCC, HCP and BCC convex hull triangulations are not unique, neither are their planar graph representations.

We will use graph isomorphism to determine possible structure matches. Two graphs $G = (V, E)$ and $H = (W, F)$ (where V and W are the graph vertices and E and F are the graph edges) are said to be *isomorphic* if there exists a bijective function $f : V \to W$ such that for all $v, w \in V$: $\{v, w\} \in E \Leftrightarrow \{f(v), f(w)\} \in F$. An *automorphism* is an isomorphism of G to itself. Many practical algorithms determine graph isomorphism by computing the *canonical form* of each graph. The canonical form of a graph G is a uniquely defined automorphism of G such that any two graphs are isomorphic if and only if their canonical forms are identical. Weinberg [19] provides a simple method for defining the canonical forms and finding the automorphisms of tri-connected planar graphs (a more readable description is given by Kukluk *et al* [20]). Weinberg's algorithm finds the canonical form by investigating both possible embeddings of a planar graph. Since we are dealing with the graphs of polyhedra in $\mathbb{R}^3$, an embedding already exists. We have therefore adapted the algorithm to investigate only a single embedding. By doing so, we restrict graph isomorphism to *orientation-preserving isomorphism*. Orientation-preserving isomorphism extends the above definition of graph isomorphism (we use the notation of Brinkmann *et al* [21]): two *embedded* graphs are said to be orientation-preservingly isomorphic if there exists a bijective function $f_e : E \to F$ such that if $\{e_1, e_2, \ldots, e_k\}$ is the set of edges incident with a vertex $v \in V$, in clockwise order, then $\{f_e(e_1), f_e(e_2), \ldots, f_e(e_k)\}$ is the set of edges incident with the vertex $f(v) \in W$, in clockwise order. An *orientation-preserving automorphism* is an orientation-preserving isomorphism of G to itself. Henceforth, when we refer to isomorphisms and automorphisms, the orientation preserving variants are implied.

Using the planar graphs we can rule out impossible structure matches: in order for a set of points to match a reference template, the planar graphs of their convex hulls must be isomorphic. Unfortunately, the converse is not true; due to the multiple triangulations of the FCC and HCP convex hulls, the planar graphs are insufficient to identify structures uniquely. There are triangulations of FCC and HCP convex hulls whose planar graphs are isomorphic, and triangulations of both which are isomorphic to the icosahedral planar graph. As such, using planar graph isomorphism alone would sometimes lead to more than one matching structure. Nevertheless, the planar graph representation gives us a set of point-to-point correspondences to investigate, one of which is the optimal correspondence. We define the optimal correspondence as the one which minimizes the RMSD between a set of points and a reference template after superposition (as defined in equation (3)). The total number of possible correspondences to investigate for each reference template is the product of the number triangulations of the convex hull, the number of automorphisms of each planar graph, and the two embeddings of each planar graph in $\mathbb{R}^3$. In practice, both the number of triangulations and automorphisms required is greatly reduced by the symmetries of each reference template (24-fold symmetry for simple cubic, BCC and FCC,

Algorithm 1. Pseudocode for determining local structure around a single atom. As argument, the procedure takes the positions of the atom and its neighbours, sorted in distance from the central atom. The algorithm returns the RMSD of the match found (or infinity if no match), the rotation matrix corresponding to the match, and the structure.

```
procedure DetermineStructure( A)
                          ▷ A is the set of positions of central atom and its nearest neighbours
                          ▷ A is ordered using Euclidean or topological ordering (see section 3)
  r* := ∞                                                          ▷ RMSD of best match
  Q* := 1                                                          ▷ Unit rotation
  S* := disordered                                                 ▷ No structure identified (yet)
  for S ∈ {SC, FCC, HCP, ICO, BCC} do
    W := ReferenceTemplate(S)
    U = {a_j ∈ A | j ⩽ |W|}                                        ▷ Select innermost atoms
    C := Conv(U)                                                   ▷ Calculate convex hull of U
    if a_1 ∉ C then                                                ▷ Convex hull must not contain central atom
      G := CanonicalForm(Graph(C))
      for each triangulation T_i of Conv(W) do
        G_ref := CanonicalForm(Graph(T_i))
        if G = G_ref then                                          ▷ Test graph isomorphism
          for each automorphism A_j of G do
            U' := U ∘ A_j                                          ▷ Permute by automorphism
            {r, Q} := RMSD(U', W)                                  ▷ Optimal superposition (see equation (3))
            if r < r* then
              r* := r
              Q* := Q
              S* := S
            end if
          end for
        end if
      end for
    end if
  end for
  return {r*, Q*, S*}
end procedure
```

6-fold symmetry for HCP, 60-fold symmetry for icosahedral); the algorithm for generating the set of symmetrically inequivalent triangulations and automorphisms is described in the appendix.

The structural identification process is described in algorithm 1. An outline of the algorithm is as follows: for each atom, we loop over the possible structures to identify. The convex hull formed by the neighbouring atoms is calculated (the number of neighbouring atoms depends on the candidate structure), and the canonical form of the corresponding graph is found. We then loop over all possible triangulations of the reference structure (skipping triangulations that are symmetrically equivalent), and calculate the corresponding canonical form. If the canonical form of the reference structure is identical to the canonical form of the actual complex hull, we have a *possible* structural match. The quality of that match is then tested. This test is done by iterating over all symmetrically inequivalent automorphisms of the candidate graph, generating all possible structural templates

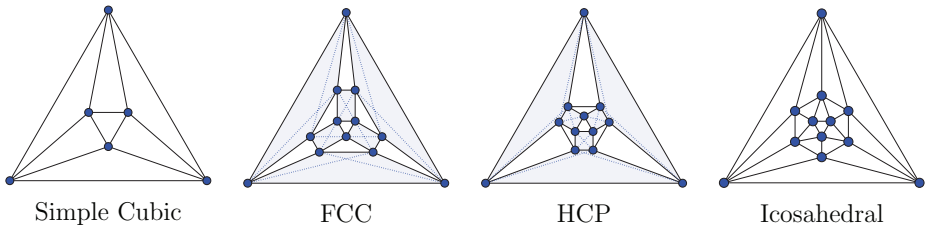


Figure 4. Convex hulls in planar graph form. The simple cubic and icosahedral graphs are unique, and are shown here as Tutte embeddings [18]. The FCC and HCP convex hulls do not have unique triangulations. The regions of the planar graphs which correspond to the square facets have been shaded. The dotted lines represent the two possible triangulations of each square facet. The BCC convex hull has no triangular facets, and cannot therefore be represented in the above manner.

which are then tested against the actual positions of the neighbouring atoms by optimizing the RMSD in equation (3). The structure with the lowest RMSD is identified, and the algorithm returns the RMSD value, the corresponding orientation and the kind of structure identified.

5. Results

5.1. Copper precipitate benchmark

To measure the capabilities of the method, we will first make a comparison with two existing structural identification methods, adaptive common neighbour analysis (ACNA) and neighbour distance analysis (NDA), both due to Stukowski [11]. Stukowski uses a Cu-rich 9R precipitate in a BCC-Fe system with 88 737 atoms in total as a benchmark to compare a number of structural classification methods. The benchmarked system, which is publicly available [22], is a case where CNA [2], bond-angle analysis [23] and voronoi-analysis perform badly, and ACNA and NDA perform very well.

We have applied our PTM algorithm to the same benchmark system. Figure 5 shows histograms of the RMSD values of the structures identified as BCC, HCP, FCC and SC. We see that a large number of atoms are classified as BCC, FCC and HCP with relatively low RMSD, these are the atoms belonging to locally crystalline areas. In addition, a much lower number of atoms are identified as HCP, FCC or SC but with a much higher RMSD. These are atoms in locally disordered structures, but where the local environment provides a poor match to one of the structural templates. One can choose to eliminate these spurious matches by introducing a cut-off, $\mathrm{RMSD}_{\mathrm{max}}$, for example based on a histogram such as figure 5.

Table 2 compares the performance of NDA and ACNA with PTM, both with no cut-off ($\mathrm{RMSD}_{\mathrm{max}} = \infty$) and with a sensible cut-off ($\mathrm{RMSD}_{\mathrm{max}} = 0.12$). PTM is capable of indexing even highly distorted structures, which is very useful at high temperatures, but a good cut-off is required to avoid spurious classifications, such as the SC and ICO classifications shown in table 2. Other than plotting a RMSD-histogram, there is no simple method for selecting $\mathrm{RMSD}_{\mathrm{max}}$, but the cut-off can fortunately be chosen *after* the analysis itself has been performed. The 'correct' value is dependent on the system being studied, as we will show in section 5.2. For this reason, future publications which use PTM should report the value of $\mathrm{RMSD}_{\mathrm{max}}$ used to ensure reproducibility.

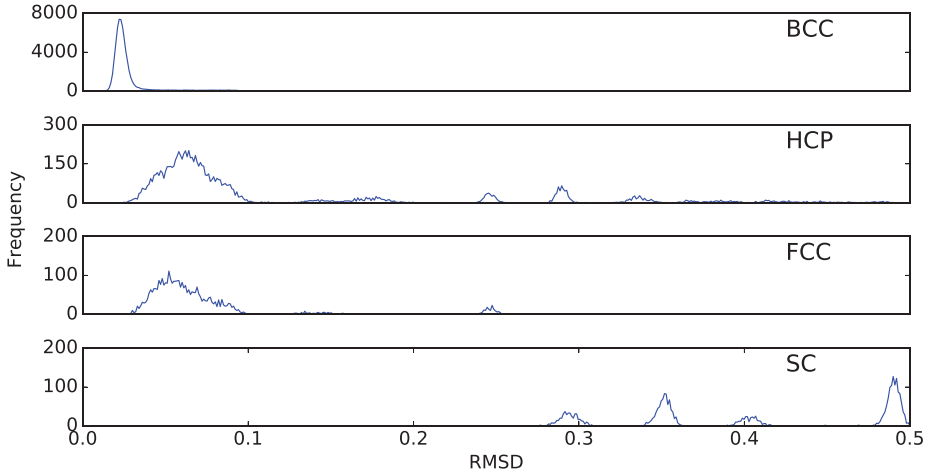


Figure 5. RMSD histograms of BCC, HCP, FCC and SC structures in a Cu-rich 9R precipitate in BCC-Fe system. The BCC atoms have very low RMSD values, which means the local structure is highly ordered. The HCP and FCC structures have both slightly higher RMSD values which are nonetheless crystalline, and very high RMSD spurious identifications. The SC identifications are exclusively spurious and are in fact highly disordered structures.

Note that the system here has been quenched, and as a consequence the local structures exhibit little distortion. In this case PTM does not provide any real improvements in structural identification compared to ACNA and NDA. The significant benefits of PTM are to be found in high-temperature systems.

5.2. Copper-Platinum alloy system at high temperature

The PTM method is very robust against thermal displacements and strain, and in high-temperature simulations it provides a significant improvement over existing identification methods. We demonstrate this here with a simulated Cu_3Pt alloy sample containing 2.8 million atoms, with periodic boundary conditions. The sample initially contains 30 grains, with randomly selected centres and orientations. The grain volumes have been constructed by computing the Voronoi cells of the grain centres. The atoms within the grains are initially ordered with a perfect $L1_2$ structure. This structure, which is the stable low-temperature phase of this system, can be considered as an FCC structure with three Cu atoms and one Pt atom in the unit cell. It should therefore be identified as FCC by the PTM algorithm.

The sample has first been quenched using the FIRE minimization method [25], and subsequently annealed at 1100 K for 820 ps using molecular dynamics with a 5 fs time step. The interatomic potential is the effective medium theory potential [26] and the temperature is controlled with a Berendsen thermostat [27]. After annealing, the sample has been separately heated to 1300 K and cooled to 900 K, 700 K and 500 K. Figure 6 compares the performance of ACNA and PTM on the Cu_3Pt systems at 500 K and 1100 K (both below the melting point temperature) and 1300 K (near the melting point temperature). Despite the system at 1300 K being in the process of melting from the grain boundaries, PTM correctly identifies the majority of FCC structures in the non-melted volumes. Topological ordering of neighbours improves

Table 2. Comparison of of PTM with two other structural analysis methods: adaptive common neighbour analysis (ACNA) and neighbour distance analysis (NDA).

Analysis method	Time (s)	Disordered	SC	FCC	HCP	ICO	BCC
ACNA[†]	0.66	12 789	0	3138	7108	0	65 702
NDA[†]	5.86	13 260	0	3453	7825	0	64 199
PTM $\mathrm{RMSD}_{\max} = 0.12$	0.82	12 900	0	3159	6971	0	65 707
PTM $\mathrm{RMSD}_{\max} = \infty$	0.82	7115	2789	3420	8970	8	66 435

Note: PTM gives similar results to both methods, though with a slightly longer running time than ACNA. The neighbours are ordered by Euclidean distance rather than topologically since the system has been quenched. A good $\mathrm{RMSD}_{\max}$ cut-off is required to avoid false positives.
[†]Both the benchmark data and the implementations of ACNA and NDA are available online [22]. Running times are for analysis only and do not include neighbour-list generation. They were measured on a 2014 MacBook Pro with an Intel Core i7-4770HQ 2.20 GHz CPU and 16 GB RAM.

detection of bulk crystallinity at high temperatures. At 1300 K, 9.6% of the atoms identified as FCC with topological ordering were identified as disordered with distance ordering.

Figure 7 shows the effect of temperature on the RMSD distribution of the structures identified as FCC. Higher temperatures lead to more distorted structures. By choosing good values for $\mathrm{RMSD}_{\max}$, spurious identifications (in this case SC, ICO and BCC) are significantly reduced. Figure 8 shows the proportion of each structure type identified for the systems shown in figure 6.

6. Alloy structures

Since PTM finds the optimal point-to-point correspondences between a set of points and an ideal reference template, we can easily identify alloy structures. A good description of the possible lattice structures of FCC and BCC alloys is given in [28]. Some figures are recreated in figure 9. Here, we will only consider binary alloys, though multi-element alloys would be a simple extension.

Identification of the A1 and A2 structures is trivial; all nearest neighbours must have the same atom type as the central atom. The B2 structure is equally simple to identify; the types of all the atoms in the first shell of neighbours must be the opposite of the central atom, and all the atoms in the second shell must have the same type as the central atom. For the $\mathrm{L1}_0$ and $\mathrm{L1}_2$ structures, it is instructive to view the structures from the central atom. This is shown in figure 10.

Using the point-to-point correspondences, the atom types of a distorted structure can be mapped onto the ideal FCC lattice structure. The alloy type can then be determined by examining the types of coplanar neighbours. Table 3 summarizes the rules for determining FCC alloy structures.

To illustrate the method described here, we have determined the alloy structures of the $\mathrm{Cu_3Pt}$ system described in section 5.2, both before annealing and after annealing. After the sample grains have been initialized by Voronoi cell construction, with perfect $\mathrm{L1}_2$ ordering, the sample has been quenched. The sample has also been quenched, post-annealing, to allow the alloy structures to be compared. The alloy structures of each system are shown in figure 11. It can be seen that grain boundary migration affects the alloy structure; despite the atoms still having FCC structure after passing through a grain boundary, the $\mathrm{L1}_2$ ordering has been lost. Before annealing, less than 0.5% of the FCC atoms were not identified as being in the $\mathrm{L1}_2$ structure. After annealing, this percentage had grown to approximately 12%.

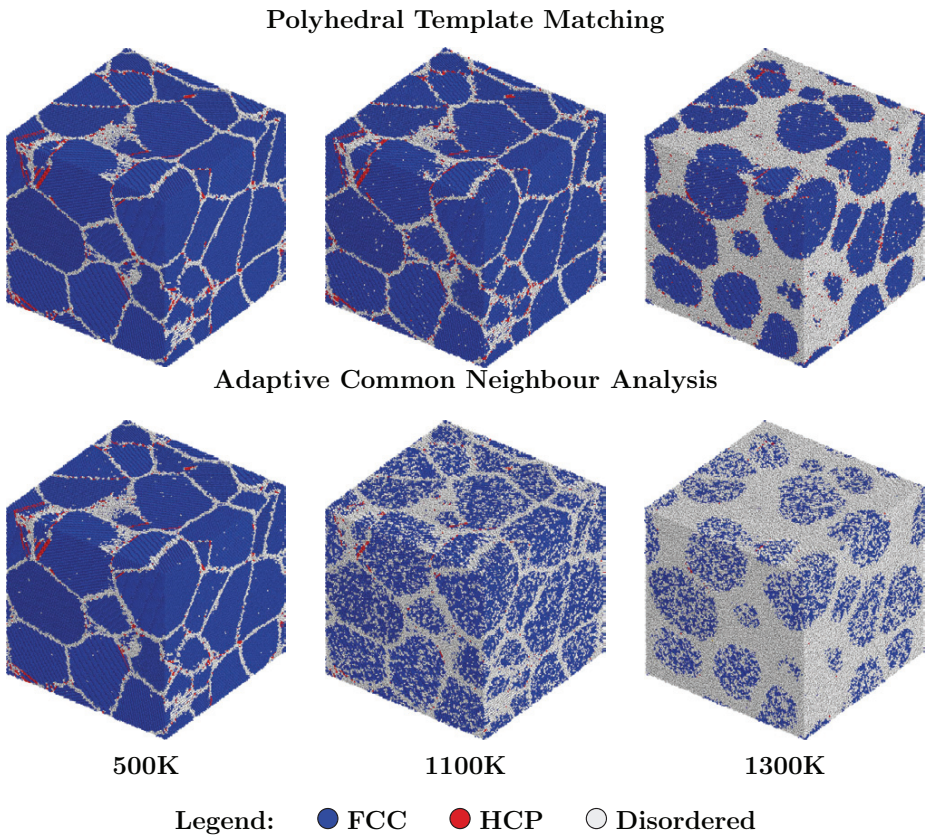


Figure 6. Comparison of PTM with ACNA at three different temperatures on Cu_3Pt systems containing 2.8 million atoms. At 500 K PTM offers little improvement over ACNA. With increasing temperature the robustness of PTM is more evident. At 1300 K the system is in the process of melting from the grain boundaries. Here, PTM finds twice the number of FCC atoms as ACNA. The $RMSD_{max}$ values used for PTM are 0.11, 0.15 and 0.17 respectively. The choice of RMSD values is motivated by the histograms in figure 7, but is not critical; using 0.17 for all figures would result in some grain boundary atoms identified as HCP. The SC, ICO and BCC classifications are shown as disordered atoms. The ACNA analyses shown here as well as all renders were performed with OVITO [24].

7. Lattice orientations

A consequence of PTM identifying structures by minimizing the RMSD (see equation (3)) is that the orientation is defined for each atom, at no extra computational cost. Furthermore, due to the optimal point-to-point correspondence being determined, the orientation is robustly determined. Figure 12 shows the lattice orientation of the FCC atoms in the Cu_3Pt system, post-annealing. The non-FCC atoms have not been included in the render.

The colours used are obtained by rotation of the orientation of each atom into its fundamental zone and projection into Rodrigues–Frank space. For materials with cubic symmetry, the fundamental zone (in Rodrigues–Frank space) is a truncated octahedron [29], which permits a

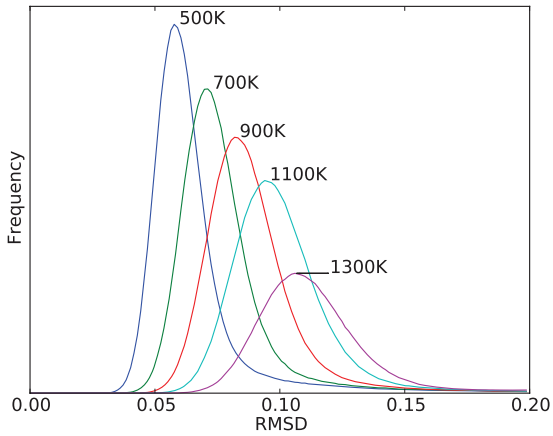


Figure 7. Histograms of RMSD values for the FCC atoms in a polycrystalline Cu_3Pt system at different temperatures. With increasing temperature the thermal displacements result in larger RMSD values.

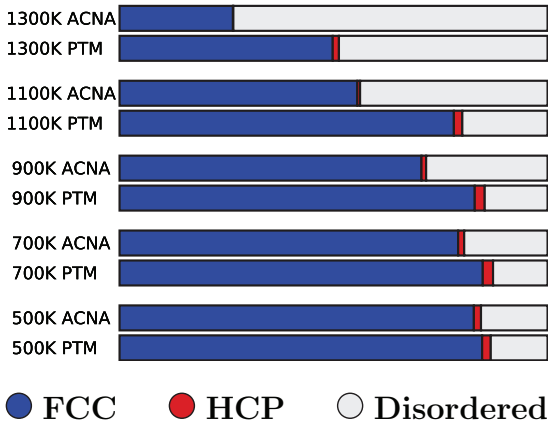
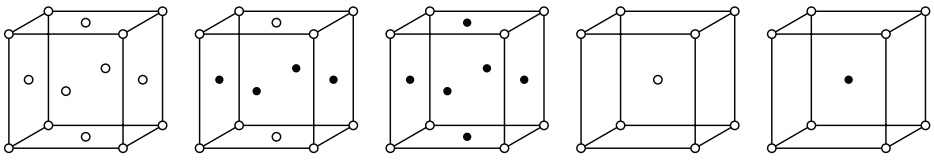
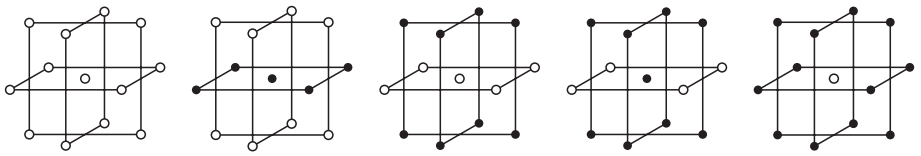


Figure 8. Relative proportions of FCC, HCP and disordered structures found with ACNA and PTM at different temperatures. The SC, ICO and BCC classifications have been counted as disordered.



(a) A1 (Cu) **(b)** $L1_0$ (CuAu) **(c)** $L1_2$ (Cu_3Au) **(d)** A2 (W) **(e)** B2 (CsCl)

Figure 9. Lattice structures of the three FCC lattice types (a)–(c), and the two BCC lattice types (d)–(e). (a) A1 (Cu). (b) $L1_0$ (CuAu). (c) $L1_2$ (Cu_3Au), (d) A2 (W), (e) B2 (CsCl).



(a) A1 Cu-type (b) $L1_0$ Cu-type (c) $L1_0$ Au-type (d) $L1_2$ Cu-type (e) $L1_2$ Au-type

Figure 10. Alloy lattice structures of a central atom and the first shell of nearest neighbours in a FCC lattice, for different central atom types. Different alloy types can be determined by counting the atom types of coplanar neighbours. (a) A1 Cu-type, (b) $L1_0$ Cu-type, (c) $L1_0$ Au-type, (d) $L1_2$ Cu-type (e) $L1_2$ Au-type.

Table 3. Rules for determining FCC binary alloy types using the types of coplanar neighbours.

Alloy structure	Atom type	Neighbours
A1	Cu	3×4 coplanar Cu-type (all Cu-type)
A1	Au	3×4 coplanar Au-type (all Au-type)
$L1_0$	Cu	2×4 coplanar Au-type, 4 coplanar Cu-type
$L1_0$	Au	2×4 coplanar Cu-type, 4 coplanar Au-type
$L1_2$	Cu	2×4 coplanar Cu-type, 4 coplanar Au-type
$L1_2$	Au	3×4 coplanar Cu-type (all Cu-type)

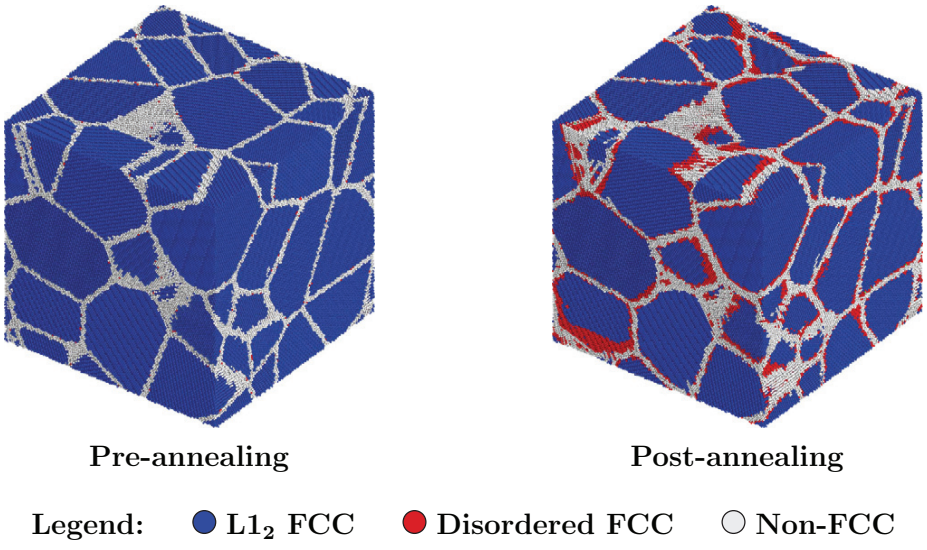


Figure 11. Alloy structures in quenched polycrystalline Cu_3Pt systems. Before annealing (left), the grains have been initialized with perfect $L1_2$ ordering. After annealing at 1100 K for 820 ps (right), the atoms which have passed through a grain boundary have disordered FCC alloy structure. The cut-off used is $RMSD_{max} = 0.05$.

Figure 12. Orientations of FCC atoms in a polycrystalline Cu_3Pt system, with $RMSD_{max} = 0.05$.

straightforward mapping into RGB colour space [30, 31]. A drawback of the colour scheme is that grains containing atoms whose fundamental orientations lie close to the faces of the fundamental zone can have very different colours. This can be seen in the blue/beige grain (lower right) in figure 12. Nevertheless, the colour scheme conveys the relationship between the grain orientations well.

8. Local elastic strain

We have the optimal point-to-point correspondences between the actual atomic positions and the positions of the ideal structure. As such, the local elastic strain is easily obtained, without any reference to an undeformed configuration at the beginning of the simulation. First, we find the deformation gradient using a least-squares fit:

$$r = \min_{\mathbf{A}} \|\mathbf{v}\mathbf{A}^T - \mathbf{w}\| \tag{6}$$

where r is the residual term, $\mathbf{v}$ and $\mathbf{w}$ are $3 \times N$ matrices containing the positions of the ideal positions and the optimally permuted actual positions respectively, and the deformation gradient, $\mathbf{A}$, is the affine transformation which minimizes the residual term. The residual term is equivalent to Falk and Langers $D^2_{\min}$ term for identifying local irreversible shear transformations [32]. Prior to fitting the deformation gradient, the ideal and actual positions are translated such that the barycentre of each set lies at the origin, and scaled according to equations (4) and (3) respectively. Although the deformation gradient obtained is scale-invariant, the scale factor can be used to recover the hydrostatic component.

The orientation and elastic strain matrices are obtained via a left-sided polar decomposition of the deformation gradient:

$$\mathbf{PU} = \mathbf{A} \tag{7}$$

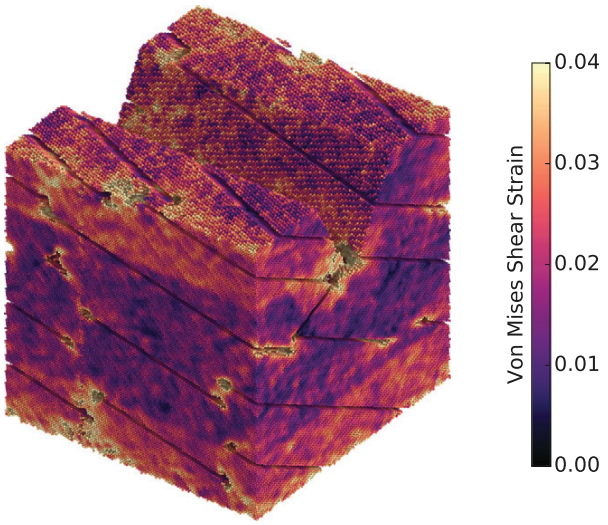


Figure 13. Local Von-Mises shear strain of FCC atoms in a single crystal $CuPt_3$ system, with $RMSD_{max} = 0.02$. Regions of elastic strain occur around the wedge tip, close to surfaces and near dislocations. Strain analysis requires quenching or time-averaging of positions, to avoid strains being dominated by atomic thermal displacements.

where **U** is an orthogonal right-handed matrix (the rotation matrix), and **P** is a symmetric matrix (the elastic strain matrix). The choice of a left-sided polar decomposition is arbitrary, but we find the elastic strain in the same frame of reference preferable for comparison of strains across different grains. In the case where **P** is not the identity matrix, **U** is not the same rotation found by minimizing the RMSD, since the addition of strain means we no longer have a rigid-body transformation. The residual term in equation (6) could be used to determine the local structure instead of the RMSD, however, the elastic strains in MD simulations are typically less than 5%, and the extra degree of freedom provided by the strain matrix often results in highly-strained spurious structural identifications.

Figure 13 shows the Von-Mises shear strain for the FCC atoms in a $CuPt_3$ system. The Von-Mises shear strain is given by:

$$\varepsilon_{\mathrm{VM}} = \sqrt{\frac{3}{2}\sum_{ij}\mathbf{P}_{ij}^2 - \frac{1}{2}\left(\sum_{k}\mathbf{P}_{kk}\right)^2} \tag{8}$$

where the sums are over the three coordinates. This is the strain equivalent of the Von Mises shear stress, defined in most solid mechanics textbooks [33]. A wedge has been cut out along the length of the system, and a 5% uniaxial strain has been applied transverse to the wedge tip direction. Periodic boundary conditions have been applied along the wedge tip and strain directions. The system was initially a single crystal (to avoid the strain field being dominated by grain boundaries) and has been annealed at 700 K for 15 000 steps of 5fs to allow stacking faults and dislocations to form. Unfortunately, either quenching or time-averaging of atomic positions is necessary for strain analysis; since the strains are very small, they are dominated by thermal displacements at even moderate temperatures. As such we have quenched the system using the FIRE [25] method. As expected, stress concentrations are seen near dislocation cores and at the tip of the wedge.

9. Implementation

The core components of the PTM algorithm (RMSD optimization, convex hull computation and graph canonization) can be computationally expensive if not implemented carefully. The RMSD optimization is solved using the library provided by Theobald [15, 34], which is much faster than SVD based calculations. Commonly used convex hull libraries, such as QHull [35] and CGAL [36], implement the QuickHull algorithm (with expected running time $O(n \log n)$) and are designed to achieve good performance on large point sets. On small point sets such as ours theoretical running times are less important to performance than efficient memory allocation, layout and accesses. As such, we have implemented a stack-allocated incremental convex hull algorithm [37] which, despite having a $O(n^2)$ theoretical running time, is 15 times faster than QHull for our point sets. For graph canonization, we have implemented a single-embedding adaptation of Weinberg's algorithm [19], again with stack allocation. Rather than representing the optimal Weinberg code as a list of edges, we have devised a hash function which uniquely represents all triconnected planar graphs as a 64-bit integer. We have ensured that no hash collisions occur by generating all possible graphs of this type (up to 15 vertices) using the `plantri` program [21]. The deformation gradients in equation (6) are solved by precomputing a Moore–Penrose pseudo-inverse matrix for each structure. This reduces solution of a least-squares problem to matrix multiplication. The polar decomposition of the deformation gradient is computed using a SVD library optimized for 3×3 matrices [38]. For the benchmark system used in section 5.1, the main components account for the following percentages of the total running time: 46% —convex hull construction, 20%—canonical form calculation, 15%—RMSD optimization (including scaling and translation of the points), and 9%—strain calculation. When using Euclidean ordering of neighbouring atoms, the method is capable of indexing over 100,000 atoms per second on a laptop computer using a single core, which is approximately 25% slower than ACNA. Topological neigbour ordering requires the computation of the Voronoi cell of each atom. To do so, we calculate the Delaunay triangulation of a central atom and its 18 nearest neighbours, which allows for up to half the inner shell atoms in FCC and HCP lattices to be wrongly ordered by Euclidean distance. We have implemented the Delaunay triangulation using the parabolic lifting map method of Edelsbrunner and Seidel [39] as it requires fewer intermediate simplices than the Bowyer–Watson [40, 41] algorithm and consequently allows for a stack-allocated implementation. Nonetheless, topological ordering requires significantly more computational effort than Euclidean ordering and increases the running time by a factor of 2. The PTM source code is available online [13].

10. Conclusions

We have presented a computationally efficient yet robust method for identifying the local structure in atomic-scale simulations. The method is based on using the topology of the convex hull formed by the neighbours of an atom to construct a small set of candidate structures, the best of which is chosen based on the root-mean-square deviation of the positions from their expected positions. In case of local crystalline symmetry, the method also identifies the local orientation of the crystalline axes, and optionally the elastic strain tensor. At low to moderate temperatures, the performance of the method is very similar to preexisting methods such as Adaptive Common Neighbour Analysis, but at higher temperatures it is more robust, and can with high reliability identify the structure of a crystalline phase for all temperatures up to the melting point.

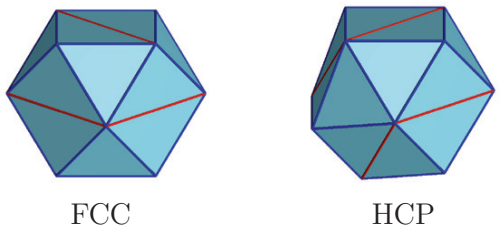


Figure A1. The two different edge types in the convex hull graphs of the FCC and HCP structures; edges which cross a square facet (red) and edges which do not (blue).

Acknowledgments

We would like to thank E A Lazar for fruitful discussions of the manuscript. We gratefully acknowledge funding from The Danish National Research Foundation's Center for Individual Nanoparticle Functionality (DNRF54).

Appendix. Symmetrically inequivalent triangulations

In section 4 we described how a convex hull can have multiple triangulations of its surface. Here we describe how the symmetrically unique triangulations can be generated, which results in a smaller number of templates to match against a structure.

The SC and ICO structures are simple cases. They both have single convex hull triangulations. Whilst the graph of their convex hulls have 24 and 60 orientation-preserving automorphisms respectively, these automorphisms are symmetrically equivalent. The FCC and HCP cases are slightly more complex. We can generate the unique triangulations by assigning colours to the edges in the convex hull graph. There are two edge lengths in the convex hull; edges which cross a square facet and edges which do not. This is shown in figure A1. The unique graphs are now the unique edge-colour-preserving and orientation-preserving isomorphisms. The symmetrically unique automorphisms can be determined either by finding the unique RMSD values, or by colouring the facets by type (equilateral triangle or half a square facet) and finding the unique orientation-preserving isomorphisms of the dual graph.

The unique BCC triangulations can be found by colouring the vertices by shell number, and finding the unique vertex-colour-preserving and orientation-preserving isomorphisms. The unique automorphisms can be found either by finding the unique RMSD values, or by colouring the facets according to the number of vertices in each shell and finding the unique orientation-preserving isomorphisms of the dual graph. Whilst we assign colours to edges and vertices during template generation, no colours are assigned during template matching. Colour assignment requires prior knowledge of facet types and, in the BCC case, the neighbour vertex shell numbers. However, the purpose of template matching is to accurately determine the structure. The lack of assumptions about facet types and shell numbers results in robust structural identifications even in highly distorted structures.

References

[1] Kelchner C L, Plimpton S J and Hamilton J C 1998 *Phys. Rev.* B **58** 11085–8
[2] Honeycutt J D and Andersen H C 1987 *J. Phys. Chem.* **91** 4950–63

[3] Faken D and Jónsson H 1994 *Comput. Mater. Sci.* **2** 279–86
[4] Schiøtz J and Jacobsen K W 2003 *Science* **301** 1357
[5] Yamakov V, Wolf D, Phillpot S R, Mukherjee A K and Gleiter H 2003 *Phil. Mag. Lett.* **83** 385–93
[6] Brandl C, Derlet P M and Van Swygenhoven H 2011 *Modelling Simul. Mater. Sci. Eng.* **19** 074005
[7] Jónsson H and Andersen H C 1988 *Phys. Rev. Lett.* **60** 2295–8
[8] Bailey N P, Schiøtz J and Jacobsen K W 2004 *Phys. Rev.* B **69** 144205
[9] Qi Y, Çağin T, Johnson W L and Goddard W A 2001 *J. Chem. Phys.* **115** 385–94
[10] Wells D M, Rossi G, Ferrando R and Palmer R E 2015 *Nanoscale* **7** 6498–503
[11] Stukowski A 2012 *Modelling Simul. Mater. Sci. Eng.* **20** 045021
[12] Lazar E A, Han J and Srolovitz D J 2015 *Proc. Natl Acad. Sci. USA* **112** E5769–76
[13] Larsen P M, Schmidt S and Schiøtz J 2016 Polyhedral Template Matching—Source Code Repository http://github.com/pmla/polyhedral-template-matching
[14] Horn B K P 1987 *J. Opt. Soc. Am.* A **4** 629–42
[15] Theobald D L 2005 *Acta Crystallogr.* A **61** 478–80
[16] Steinitz E 1916 *Polyeder und Raumeinteilungen* (Leipzig: Teubner)
[17] Whitney H 1933 *Am. J. Math.* **55** 231
[18] Tutte W T 1963 *Proc. Lond. Math. Soc.* **s3-13** 743–67
[19] Weinberg L 1966 *IEEE Trans. Circuit Theory* **13** 142–8
[20] Kukluk J P, Holder L B and Cook D J 2004 *J. Graph Algorithms Appl.* **8** 313–56
[21] Brinkmann G and McKay B D 2007 *MATCH Commun. Math. Comput. Chem.* **58** 323–57
[22] Stukowski A 2012 Atomic structure analysis algorithms http://asa.ovito.org
[23] Ackland G J and Jones A P 2006 *Phys. Rev.* B **73** 054104
[24] Stukowski A 2010 *Modelling Simul. Mater. Sci. Eng.* **18** 015012
[25] Bitzek E, Koskinen P, Gähler F, Moseler M and Gumbsch P 2006 *Phys. Rev. Lett.* **97** 170201
[26] Jacobsen K W, Stoltze P and Nørskov J K 1996 *Surf. Sci.* **366** 394–402
[27] Berendsen H J C, Postma J P M, van Gunsteren W F, DiNola A and Haak J R 1984 *J. Chem. Phys.* **81** 3684–90
[28] Laughlin D E 1988 *Encyclopedia of Materials Science and Engineering* vol 1 (Oxford: Pergamon) pp 263–8
[29] He Y and Jonas J J 2008 *Applications of Texture Analysis Ceramic Transactions* (Hoboken, NJ: Wiley) pp 269–80
[30] Albou A, Driver J H and Maurice C 2010 *Acta Mater.* **58** 3022–34
[31] Van Boxel S, Seefeldt M, Verlinden B and van Houtte P 2005 *J. Microsc.* **218** 104–14
[32] Falk M L and Langer J S 1998 *Phys. Rev.* E **57** 7192–205
[33] Budynas R G and Nisbett J K 2011 *Shigley's Mechanical Engineering Design (McGraw-Hill Series in Mechanical Engineering)* (New York: McGraw-Hill)
[34] Liu P, Agrafiotis D K and Theobald D L 2010 *J. Comput. Chem.* **31** 1561–3
[35] Barber C B, Dobkin D P and Huhdanpaa H 1996 *ACM Trans. Math. Softw.* **22** 469–83
[36] Hert S and Schirra S 2013 CGAL User and Reference Manual (CGAL Editorial Board) http://doc.cgal.org/4.2/CGAL.CGAL/html/packages.html#PkgConvexHull3Summary
[37] O'Rourke J 1998 *Computational Geometry in* C (Cambridge: Cambridge University Press)
[38] McAdams A, Selle A, Tamstorf R, Teran J and Sifakis E 2011 Computing the singular value decomposition of 3×3 matrices with minimal branching and elementary floating point operations *Technical Report* No. 1690 University of Wisconsin-Madison
[39] Edelsbrunner H and Seidel R 1986 *Discrete Comput. Geom.* **1** 25–44
[40] Bowyer A 1981 *Comput. J.* **24** 162–6
[41] Watson D F 1981 *Comput. J.* **24** 167–72

Paper 2

Determination of low-strain interfaces via geometric matching
Jelver L, Larsen P M, Stradi D, Stokbro K and Jacobsen K W
Physical Review B
Volume 96, Issue 8, Pages 085306

Determination of low-strain interfaces via geometric matching

Line Jelver,[1,2] Peter Mahler Larsen,[2] Daniele Stradi,[1] Kurt Stokbro,[1] and Karsten Wedel Jacobsen[2,*]
[1]*QuantumWise A/S, Fruebjergvej 3, DK-2100 Copenhagen, Denmark*
[2]*Department of Physics, Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark*
(Received 10 May 2017; published 22 August 2017)

We present a general method for combining two crystals into an interface. The method finds all possible interfaces between the crystals with small coincidence cells and identifies the strain and area of the corresponding two-dimensional cells of the two crystal surfaces. We apply the method to the two semiconductor alloys $InAs_{1-x}Sb_x$ and $Ga_xIn_{1-x}As$ combined with a selection of pure metals or with NbTiN to create semiconductor/superconductor interfaces. The lattice constant of the alloy can be tuned by composition and we can extract the alloy lattice parameters corresponding to zero strain in both the metal and the alloy. The results can be used to suggest new epitaxially matched interfaces between two materials.

DOI: 10.1103/PhysRevB.96.085306

I. INTRODUCTION

The development of modern technology has become increasingly dependent on knowledge of interfaces at the atomic scale. As the size of electronic devices decreases, interfaces become an increasingly dominant part of the system and thus become the limiting factor for device performance [1–3]. Many difficulties are thus related to obtaining a stable and defect-free interface. When the materials are not commensurate, a large strain can build up at the interface, and result in defects and unstable interface geometries. To this end, it can be beneficial to use an alloy as one of the interface materials. By changing the alloy composition, the lattice constant for one of the interface materials can be tuned, thereby obtaining commensurate lattices across the interface. Many future devices can therefore be expected to be designed from alloys. A growing field, where alloys are commonly used, is the fabrication of core-shell nanowires. These nanowires have numerous applications, e.g., for photodetectors [4], photoelectrodes [5], and thermoelectric devices [6].

The prediction of the stability of an interface is difficult [7] and the commonly used approach is trial and error where many samples must be grown before it can be concluded whether a stable interface can be formed or not. In this paper, we present a crystal matching method which permits the combination of any two crystals and provides information on the crystal surfaces, if any, that allow for a low-strain epitaxial interface. The method is solely based on geometrical considerations of the possible surface cells of the two crystals and it leads to an identification of interfaces where both the strain and the size of the coincidence interface cell are small. Having low stress and a small interface cell does not by itself guarantee a stable interface, as the atomic structure of the interface may also play an important role. However, the simple geometrical criteria provide a good starting point for further experimental or theoretical investigations.

In a previous article [8], some of the authors have shown how to find good matches between specific surfaces of two crystals and a related, but more simplistic, method for such a two-dimensional (2D) match has also previously been proposed [9]. With the present method, all possible crystal orientations and surfaces are investigated at the same time. Furthermore, we introduce a scaling parameter between the two crystals which can be tuned to gradually change the size of one of the crystal structures while keeping the other fixed. The scaling parameter mimics the situation where the lattice constant of one of the materials can be tuned by modifying the alloy composition, and it provides a convenient parameter for analyzing and understanding the interface-matching problem. We derive an analytical relation between the scaling parameter and the minimal strain, and demonstrate its usefulness in Sec. IV, where we investigate the matches between an arbitrary fcc crystal and an arbitrary bcc crystal.

The introduction of the scaling parameter makes the method an ideal tool for investigating interfaces containing alloys. In Sec. V, we apply the method to the two semiconductor alloys $InAs_{1-x}Sb_x$ and $Ga_xIn_{1-x}As$ combined with a range of metals (Al, Ni, Cu, Ag, Au, Pb, V, Fe, Nb, and Co) and for each metal we predict the alloy lattice constants which lead to strain free interfaces. InAs and InSb are, for example, used in semiconductor-superconductor core-shell nanowires for the realization of zero-energy localized Majorana modes [10–15]. The results obtained here therefore provide some guidelines for promising new combinations of semiconductor alloy compositions and metals for core-shell nanowires. Promising results for the generation of Majorana modes have also been shown for an InSb-$Nb_{1-x}Ti_xN$ interface [16]. The final part of this work is to investigate how the two semiconductor alloys match with this superconducting alloy.

II. INTERFACE ENERGETICS

The method we are going to describe below makes it possible to identify interfaces with small coincidence cells where only a small strain is required. The model thus takes only the material lattices into account, but not the atomic details. To what extent such interfaces will be stable is highly dependent on the particular materials and the character of the bonding. To discuss this we consider the situation where two materials A and B are homogeneously strained and put together to form an interface. The energy difference ΔE associated with straining the materials and forming the interface can be divided into

*kwj@fysik.dtu.dk

three contributions,

$$\Delta E = \Delta E^{\text{int}} + \Delta E^{\text{surf. strain}} + \Delta E^{\text{bulk strain}}, \quad (1)$$

where the interface term is the energy gained by forming the interface from the surfaces at fixed strain,

$$\Delta E^{\text{int}} = E^{\text{int}}_{\text{AB}}(\epsilon_A,\epsilon_B) - E^{\text{surf}}_{A}(\epsilon_A) - E^{\text{surf}}_{B}(\epsilon_B). \quad (2)$$

The second term is the energy change in the surface energy because of the strain,

$$\Delta E^{\text{surf. strain}} = E^{\text{surf}}_{A}(\epsilon_A) - E^{\text{surf}}_{A}(\epsilon_A = 0) + E^{\text{surf}}_{B}(\epsilon_B) - E^{\text{surf}}_{B}(\epsilon_B = 0). \quad (3)$$

This contribution can be both positive or negative and we shall in the following assume that this term can be considered small. The third term is the energy cost associated with straining the bulk of the materials,

$$\Delta E^{\text{bulk strain}} = E^{\text{bulk}}_{A}(\epsilon_A) - E^{\text{bulk}}_{A}(\epsilon_A = 0) + E^{\text{bulk}}_{B}(\epsilon_B) - E^{\text{bulk}}_{B}(\epsilon_B = 0). \quad (4)$$

If we assume that the dependency of the surface energy with strain can be neglected, we expect to get a stable interface if the energy gain from the interface ΔE^{int} dominates the cost from straining the bulk of the materials $\Delta E^{\text{bulk strain}}$. The model suggested here identifies interface matches with low-strain and small interface coincidence cells. The small strain will clearly tend to minimize the bulk strain energy as this scales quadratically with the strain. A more questionable assumption is that a small interface cell will lead to strong bonding at the interface. The bonding will often depend quite sensitively on the atomic structure at the interface, which is not considered at all by the model. However, in many cases where the interface cell is small, a translation of one of the surfaces relative to the other makes it possible to obtain favorable bonding configurations which are then repeated over the entire interface, leading to general high stability. However, if the interface cell is large, the atomic bonding configurations will often vary considerably over the cell, so that in some parts of the cell favorable bonding configurations are obtained but in other parts not. Overall this leads to weaker bonding [17].

Another advantage of a small coincidence cell is that it may be more stable with respect to shear in the interface. For a small cell, a large corrugation in the energy landscape as a function of displacement of one of the materials relative to the other one along the interface can be expected. The bonds across the interface will respond to the shear "in synchrony" leading to large variation. For a larger cell where some bonds are strong and others are weak, the bonds will respond differently to the shear presumably leading to a smaller energy corrugation.

If the materials are thick, the strain has to be very small since the bulk strain energy grows in proportion to the thickness. In this limit, only interface matches at very low strain will be acceptable. However, in practice this limit may also lead to incommensurate interfaces or defects at or close to the interface, situations clearly beyond what a simple lattice model can account for. Similarly, if the interface bonding energy is very small and has a small corrugation, as is, for example, the case with van der Waals bonding, stable interfaces with large moiré-pattern coincidence cells or even incommensurate cells may occur. This situation can arise even for very thin films as, for example, in the case of graphene on some metal surfaces, where the interaction is weak. Despite the fact that a graphene layer is atomically thin, the strong interatomic bonding within the graphene layer results, for some metals, in little accommodation of the surface and large coincidence cells as a result [18,19].

III. CRYSTAL MATCHING METHOD

Our method for creating an interface between two crystals is general and based on 2D cells of the two crystal surfaces. The 3D vectors of the crystal, defining these surface cells, are projected from $\mathbb{R}^3$ to $\mathbb{R}^2$ as illustrated in Fig. 1(a). We begin by explaining the method behind matching two such cells and extracting the related strain.

Let the surface cell of the first crystal be defined by two vectors $\mathbf{u}_1$ and $\mathbf{u}_2$, where $\mathbf{u}_1 = [u_{1x}, u_{1y}]^T$, as shown in Fig. 1(a). Similarly, let $\mathbf{v}_1$ and $\mathbf{v}_2$ denote the two vectors which define the surface cell of the second crystal. Then, the affine transformation $\mathbf{A}$ which maps $[\mathbf{u}_1,\mathbf{u}_2]$ onto $[\mathbf{v}_1,\mathbf{v}_2]$ is given by

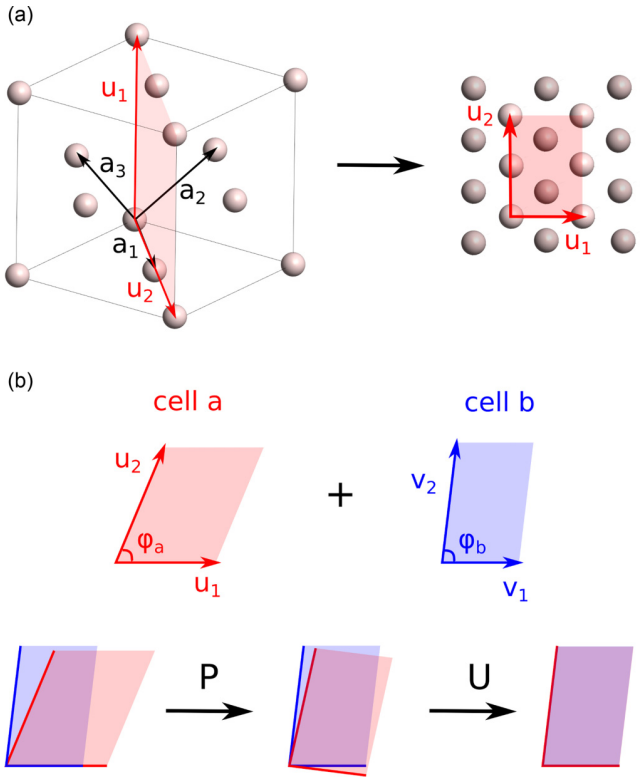


FIG. 1. The method for finding and matching the 2D surface cells of two crystals. (a) The lattice vectors of the first crystal $(\mathbf{u}_1,\mathbf{u}_2)$ are created from a linear combination of the Bravais vectors $(\mathbf{a}_1,\mathbf{a}_2,\mathbf{a}_3)$. Here $\mathbf{u}_1 = -\mathbf{a}_1 + \mathbf{a}_2 + \mathbf{a}_3$ and $\mathbf{u}_2 = 2\mathbf{a}_1$. The two vectors are then projected from the three-dimensional (3D) representation to a 2D representation on a crystal surface. (b) Two cells of two different crystals are matched by applying the affine transformation $\mathbf{A}[\mathbf{u}_1,\mathbf{u}_2] = [\mathbf{v}_1,\mathbf{v}_2]$, where $\mathbf{A} = \mathbf{UP}$ consists of a rotation $\mathbf{U}$, and a strain matrix $\mathbf{P}$.

the following system of linear equations,

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} u_{1x} & u_{2x} \\ u_{1y} & u_{2y} \end{bmatrix} = \begin{bmatrix} v_{1x} & v_{2x} \\ v_{1y} & v_{2y} \end{bmatrix}. \tag{5}$$

Any square matrix can be decomposed into the product of an orthonormal matrix $\mathbf{U}$, and a positive definite symmetric matrix $\mathbf{P}$. This is known as a polar decomposition [20]. The symmetric matrix defines the 2D strain tensor $\boldsymbol{\epsilon}$ for deforming one cell into the other,

$$\mathbf{P} = \mathbf{I} + \boldsymbol{\epsilon} = \begin{bmatrix} 1+\epsilon_{xx} & \epsilon_{xy} \\ \epsilon_{xy} & 1+\epsilon_{yy} \end{bmatrix}. \tag{6}$$

The first vectors are rotated along the x axis s.t. $u_{1y} = v_{1y} = 0$. This can be done without loss of generality and leads to

$$A_{11} = \frac{v_{1x}}{u_{1x}}, \tag{7}$$

$$A_{12} = \frac{v_{2x}}{u_{2y}} - \frac{v_{1x}u_{2x}}{u_{1x}u_{2y}}, \tag{8}$$

$$A_{21} = 0, \tag{9}$$

$$A_{22} = \frac{v_{2y}}{u_{2y}}. \tag{10}$$

We now make the polar decomposition of $\mathbf{A}$ s.t. $\mathbf{A} = \mathbf{UP}$, where $\mathbf{U}$ is a rotation matrix because of the chosen projection to $\mathbb{R}^2$,

$$\mathbf{U} = s\begin{bmatrix} A_{11}+A_{22} & A_{12} \\ -A_{12} & A_{11}+A_{22} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}, \tag{11}$$

$$\mathbf{P} = \mathbf{U}^T\mathbf{A}, \tag{12}$$

where s is a factor which makes the columns of $\mathbf{U}$ unit vectors. The $\mathbf{U}$ matrix defines the counterclockwise rotation of the $[\mathbf{u}_1,\mathbf{u}_2]$ cell onto the $[\mathbf{v}_1,\mathbf{v}_2]$ cell by the angle $\phi = |\phi_a - \phi_b|/2$ as shown in Fig. 1(b). Using this method, Eqs. (7)–(12) thus yield the strain matrix of any given cell combination.

A. Algorithm

We now explain the algorithm behind extracting the strain matrix of all the possible matches between two crystals. The procedure is illustrated in the flow chart in Fig. 2.

1. *Create (i,j,k) list from $\ell_{\max}$*

The first step is to create all the possible lattice vectors of each crystal up to a specified maximum length $\ell_{\max}$. The vectors are created as integer combinations of the Bravais vectors of the crystal, as illustrated in Fig. 1(a),

$$\mathbf{u} = i\mathbf{a}_1 + j\mathbf{a}_2 + k\mathbf{a}_3, \quad |\mathbf{u}| < \ell_{\max}. \tag{13}$$

This will create a list of (i,j,k) values for each crystal.

2. *Create vector pair list*

The next step is to combine the created vectors such that a list of unique surface cells is created for each crystal. This procedure uses two kinds of filters to remove equivalent surface cells from the list; the symmetry operations of the atom-free crystal [21] and Niggli reductions of the cells [22].

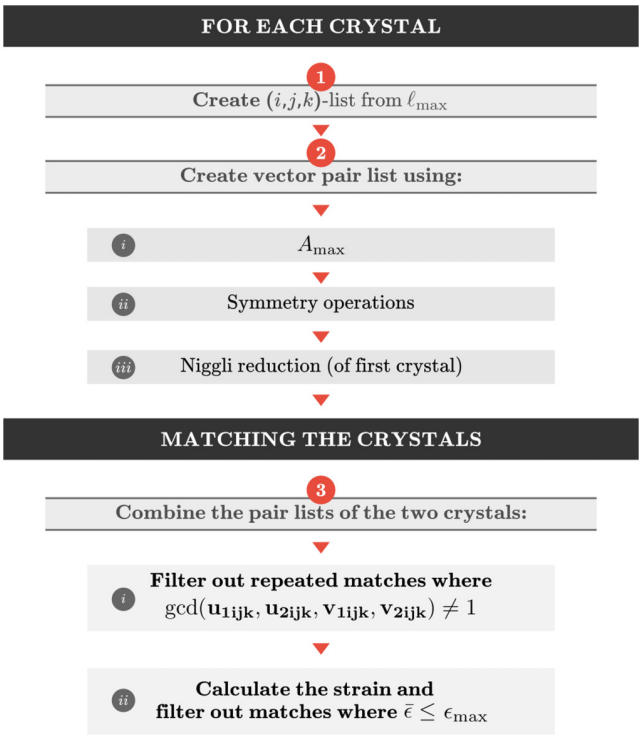


FIG. 2. Flow chart behind the algorithm for matching two crystals.

As a starting point, we create a list of symmetrically unique vectors of the crystal by applying the symmetry operations $\{\mathbf{S}\}$ to each vector in the generated (i,j,k) list. Let $\mathbf{u}$ be the vector we apply the operations to and let $\{\mathbf{Su}\}$ be the set of all the vectors created when applying the symmetry operations. We define the canonical representation of this set as its *least element* with respect to the lexicographic ordering. When iterating through the (i,j,k) list, noncanonical vectors are discarded.

To create the unique surface cells, we combine each vector from the original (i,j,k) list with the vectors in the symmetry reduced (i,j,k) list. The combination is discarded if the two vectors are parallel or if the area that they span exceeds $A_{\max}$. The symmetry operations of the crystal are then reapplied, this time on both vectors: $\{\mathbf{S}[\mathbf{u}_1,\mathbf{u}_2]\}$. Again, any noncanonical pairs are discarded.

The final test is to investigate whether the cell is a Niggli reduced cell. A Niggli reduced 2D cell fulfils

$$\mathbf{u}_1 \cdot \mathbf{u}_1 \leqslant \mathbf{u}_2 \cdot \mathbf{u}_2, \quad \mathbf{u}_1 \cdot \mathbf{u}_2 \leqslant \tfrac{1}{2}\mathbf{u}_1 \cdot \mathbf{u}_1. \tag{14}$$

The cells that are not Niggli reduced are discarded. This is only done for the first crystal, since a Niggli reduced cell may be strained into a non-Niggli reduced cell when the two crystals are combined. After these calculations, we have created a list of surface cells for each crystal and the next step is to combine these two lists.

3. *Combine the pair lists of the two crystals*

When the cells of the two crystals are matched, the first step is to filter out repeated matches. As an example, let a $[\mathbf{u}_1,\mathbf{u}_2]$ cell of the first crystal be combined with a $[\mathbf{v}_1,\mathbf{v}_2]$ cell where $\mathbf{u}_{1ijk} = (1,0,0)$, $\mathbf{u}_{2ijk} = (2,0,1)$, $\mathbf{v}_{1ijk} = (0,0,2)$, and $\mathbf{u}_{2ijk} = (4,4,2)$. This combination is equivalent to the ((2,0,0),(4,0,2),(0,0,4),(8,8,4)) combination. We avoid investigating both combinations by discarding any combinations whose greatest common divisor, $\gcd(\mathbf{u}_{1ijk},\mathbf{u}_{2ijk},\mathbf{v}_{1ijk},\mathbf{v}_{2ijk})$, is not 1. After this preliminary test, the strain matrix of the match is calculated using the method explained in the beginning of this section.

We now define a measure for the average strain of a match,

$$\bar{\epsilon} = \sqrt{\frac{\epsilon_{xx}^2 + \epsilon_{yy}^2 + \epsilon_{xx}\epsilon_{yy} + \epsilon_{xy}^2}{4}}, \tag{15}$$

where ϵ_{xx},ϵ_{xy}, and ϵ_{yy} are the components of the 2D strain tensor shown in Eq. (6). This average strain is an invariant of the strain tensor, since $4\bar{\epsilon}^2 = \mathrm{Tr}(\boldsymbol{\epsilon})^2 - \det(\boldsymbol{\epsilon})$. Matches with an average strain below a given strain threshold $\varepsilon_{\max}$ are kept.

This concludes the algorithm for finding all the matches between two crystals. The parameters determining which matches to include in the search are $\ell_{\max}$, $A_{\max}$, and $\varepsilon_{\max}$. These parameters help to filter out the cells that wouldn't create a physically meaningful interface. The area and length threshold ensures that we don't investigate unreasonably large or narrow cells and the strain threshold filters out the most strained matches.

The algorithm is implemented using C++ with a Python interface and has good performance; finding all matches between a InAs fcc crystal and a Cobalt hcp crystal with the parameters, $\ell_{\max} = 50$ Å, $A_{\max} = 200$ Å^2, and $\epsilon_{\max} = 2\%$, takes approximately 20 min on a normal laptop. The algorithm is available in VIRTUAL NANOLAB version ATK-VNL-2017 [23].

IV. MATCHING OF A FCC CRYSTAL WITH A BCC CRYSTAL

In the previous section, we have explained how to match two specific crystals with fixed lattice parameters. Here, we discuss the matching of any fcc crystal with any bcc crystal. To this end, we introduce an isotropic scaling parameter k, which is applied to the Bravais vectors of the bcc crystal. This scaling parameter can then be defined as the ratio between the lattice constants of the two crystals, $k = a_{\mathrm{fcc}}/a_{\mathrm{bcc}}$. The effect of k on the strain matrix is linear,

$$\mathbf{A} = k\mathbf{U}\mathbf{P} = \mathbf{U}\begin{bmatrix} k(1+\epsilon_{xx}) & k\epsilon_{xy} \\ k\epsilon_{xy} & k(1+\epsilon_{yy}) \end{bmatrix}. \tag{16}$$

This means that the effect of the scaling parameter on the average strain can be described by the simple analytical relation,

$$4\bar{\varepsilon}^2(k) = k^2\left(\epsilon_{xx}^2 + \epsilon_{yy}^2 + \epsilon_{xx}\epsilon_{yy} + \epsilon_{xy}^2\right) + k^2(3\epsilon_{xx} + 3\epsilon_{yy} + 3) - k(3\epsilon_{xx} + 3\epsilon_{yy} + 6) + 3, \tag{17}$$

where the strain tensor components are referring to the case of $k = 1$. This relation allows us to calculate the average strain of a match at any k value once the strain matrix has been calculated for any one specific value of k. Equation (17) defines a strain parabola; the minimal strain of a match, along with the corresponding k value, can be found from the minimum of this parabola.

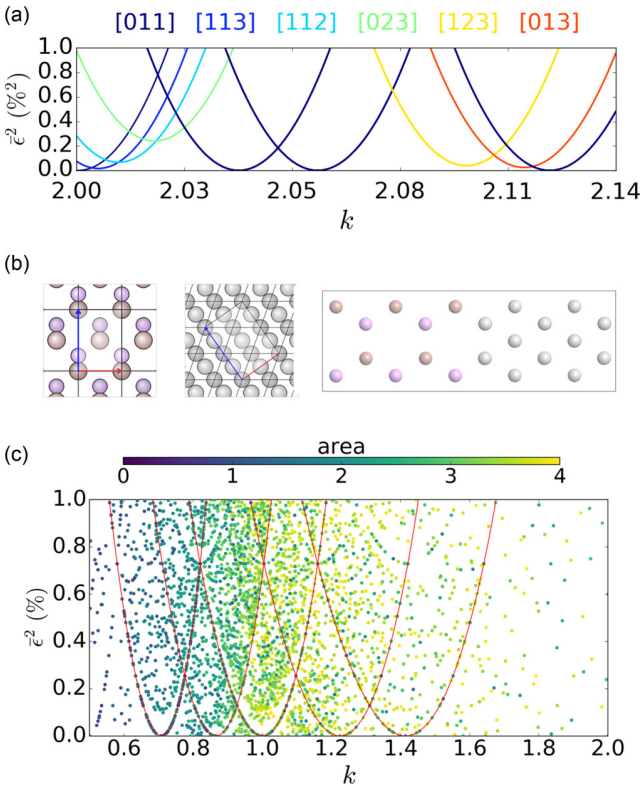


FIG. 3. The matches between a fcc and a bcc crystal. k represents the scaling between the two lattice constants, $k = a_{\mathrm{fcc}}/a_{\mathrm{bcc}}$. (a) The $\bar{\varepsilon}^2(k)$ relation of (17) for optimal matches involving the [110] surface of the fcc crystal. The color of the curves represents different surfaces of the bcc crystal. (b) The surface cells and corresponding interface of the perfect $k = 2$ match shown on (a). Illustrated here for InAs with vanadium. (c) Scatter plot of matches where the minima of the strain parabola gets below 10%. The dots represent the minimal strain of the match and the k value where this strain occurs. We have chosen $\ell_{\max}$ and $A_{\max}$ as four times the lattice constant of the unstrained crystal. The color represents the area of the fcc surface cell of the match.

The strain parabolas for the optimal matches [24] between an fcc[110] surface and a bcc crystal with k values between 2.0 and 2.14 are illustrated in Fig. 3(a). This range is of relevance for the matching of a InAsSb alloy with vanadium. It can be seen that the two [110] surfaces have perfect matches at $k = 2.0$, 2.04, 2.06, and 2.12. The surface cells and corresponding interface of the perfect $k = 2.0$ match are illustrated in Fig. 3(b) for InAs and vanadium. This k value can be obtained, when matching, e.g., $InAs_{0.996}Sb_{0.004}$ to vanadium or $Ga_{0.786}In_{0.214}As$ to iron. As such, this specific match reappears in the results of the next section where these two alloys are matched to a range of metals.

To obtain the full picture of the matches between a fcc and a bcc crystal, we use the algorithm described in Sec. III and apply the strain threshold to the minimal strain of a given match. It corresponds to the following flow chart in Fig. 2 but where step 3.ii. is altered such that after calculating the strain

matrix, we use Eq. (17) to find the minimal strain of the match and only keep the matches where this is lower than the strain threshold. In this manner, we can retrieve the strain parabolas of all the relevant matches by doing a single calculation at $k = 1$. We have chosen $\ell_{\max}$ and $A_{\max}$ as four times the lattice constant of the unstrained crystal and $\epsilon_{\max} = 10\%$.

Figure 3(c) shows the results of this general investigation. For each match, we plot the minimal strain and the corresponding k value. Note, that the y axis shows the squared strain, since this illustrates the general shape of the matches better. The structure shows curves where zero strain minimas appear, e.g., at the points $k = 1/\sqrt{2} \approx 0.71$, $k = \sqrt{3}/2 \approx 0.87$, $k = 1$, $k = \sqrt{3/2} \approx 1.22$, and $k = \sqrt{2} \approx 1.41$. The matches lying on these curves are related to a uniaxial strain between the two surface cells. The perfect matches at $k = 1$ appear since this value corresponds to the [100] facets of the two cubic crystals fitting perfectly together. All other matches on this $k = 1$ curve correspond to a match where one of the cells is rotated and then stretched in one direction only. The other zero strain points also represent some symmetry of the two crystals. The zero strain match at $k = 1/\sqrt{2}$, e.g., corresponds to matching the rotated bcc[100] surface with cell vectors $\mathbf{v}_1 = [\sqrt{2}a_{\text{bcc}}, 0]$ and $\mathbf{v}_2 = [0, \sqrt{2}a_{\text{bcc}}]$ to a [100] facet of the fcc crystal. The other matches lying on this curve represents a uniaxial strain on top of this perfect match where one of the cells is rotated, stretched in both directions by $\sqrt{2}$, and then stretched in one direction afterwards. The matches lying on a specific curve are thus related by having a fixed eigenvalue of the **P** matrix of $1/k_{\text{zsm}}$, where k_{zsm} is the k value at the zero strain minimum of the curve. For example, all matches on the $k = 1/\sqrt{2}$ curve have **P** matrices with the eigenvalue $\sqrt{2}$.

Two things are important to note about this plot. First, the length and area limits of the vectors and cells determine the density of the found matches. Without these limits the entire $(\bar{\epsilon}, k)$ space would be filled with points. Secondly, the constraining to cubic structures lead to points with nonzero strain; if the crystal structures were allowed to vary arbitrarily, all points would have zero strain.

These results demonstrate that the method represents a general tool for crystal matching. The strength of the method is that it only relies on the geometry of the crystals. This makes it possible to calculate results for two arbitrary crystals and apply these results to all interfaces between materials of these crystal structures. Furthermore, it is an ideal tool for investigating alloy crystals where the lattice parameter can be varied as the composition of the alloy is changed. This will be the subject of the next section.

V. INTERFACES BETWEEN SEMICONDUCTOR ALLOYS AND METALS

We apply the method to two different semiconductor alloys, $InAs_{1-x}Sb_x$ and $Ga_xIn_{1-x}As$, and match their surfaces with those of 10 different metals (Al, Ni, Cu, Ag, Au, Pb, V, Fe, Nb, and Co). In particular, we study how the orientation and strain of the metallic surface depends on the lattice parameter of the semiconductor surface, which can be tuned by changing the mole fraction. The two alloys form a zincblende crystal

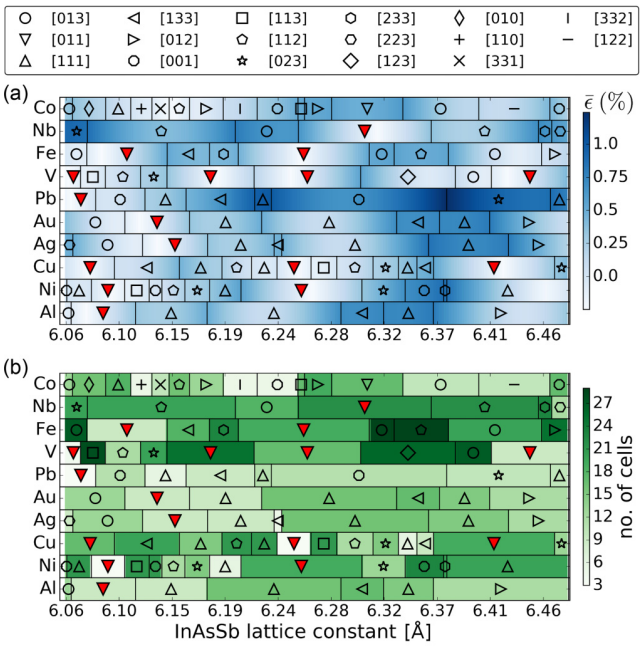


FIG. 4. (a) The InAsSb[$1\bar{1}0$] surface matched with 10 different metals. Each block represents the match which results in the lowest average strain between the two materials. The background color shows the strain in percent and the markers show the involved metal surface. Filled markers denote a match where zero strain can be obtained. (b) Same as (a) but where the background color shows the area given by the number of alloy surface unit cells of the match.

and the relation between the lattice constant and mole fraction x can be approximated by the linear relations [25]:

$$a_{\text{InAsSb}}(x) = 6.0583 + 0.4207x, \tag{18}$$

$$a_{\text{GaInAs}}(x) = 6.0583 - 0.405x. \tag{19}$$

We use the experimentally determined lattice constants [26] of the metals and strain the metal surfaces to match the alloy surfaces. The chosen parameters [27] are $\ell_{\max} = 50$ Å, $A_{\max} = 200$ Å^2, and $\epsilon_{\max} = 2\%$. In addition, we set a limit on the Miller index of the crystal surfaces. If the highest value in the Miller index is above the threshold, $m_{\max} = 3$, the match is discarded. The matches are calculated for a single value of the mole fraction and the scaling relation (17) of Sec. IV is then used to get the results for the rest of the x values. This is possible, since each x value directly corresponds to a k value, $k = a_{\text{alloy}}(x)/a_{\text{metal}}$.

In Fig. 4, we show the matches involving the [$1\bar{1}0$] surface of InAsSb. Each block in the plot corresponds to a certain match and a certain strain parabola in a plot like Fig. 3(a). The background of Fig. 4(a) therefore represents the strain value of the lowest lying strain parabola. Figure 3(a) shows InAsSb[$1\bar{1}0$] matched to vanadium and it is seen how the variation of the lattice parameter results in different optimal matches. For instance, from a perfect [110] match to a low-strain [113] match as the lattice constant is increased from 6.06 to 6.08.

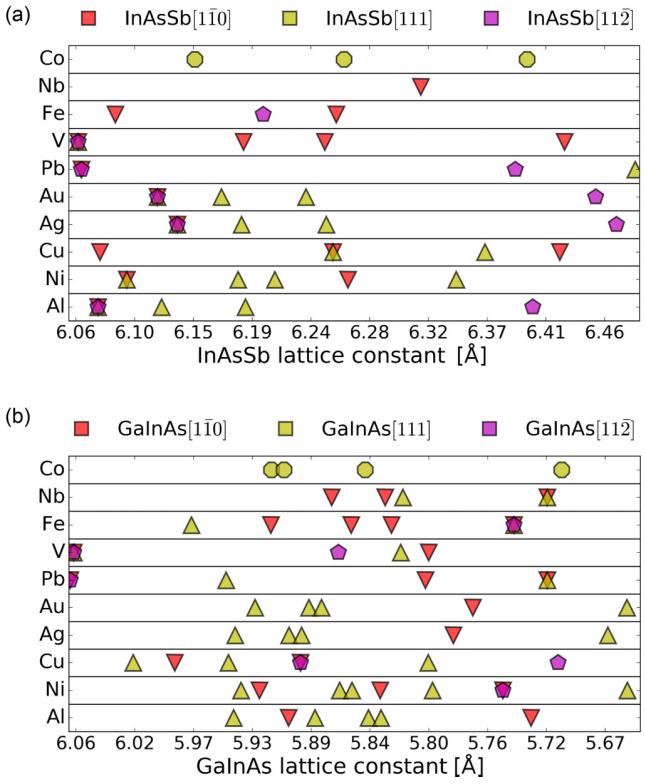


FIG. 5. The perfect matches of the [1$\bar{1}$0], [111], and [11$\bar{2}$] surfaces of (a) InAsSb and (b) GaInAs. The markers denote the metal surface (see Fig 4 for labels) and the color denotes the alloy surface. Each marker is placed at the alloy lattice constant which results in a zero strain match.

The area of the match, given as the number of alloy surface unit cells, is illustrated in Fig. 4(b). It is seen that several zero strain matches are possible between the InAsSb[1$\bar{1}$0] surface and those of the metals. Nickel, copper, lead, and vanadium even have perfect matches with very small unit cells, indicating that these interfaces will potentially be stable. The geometry of this small area match between InAsSb and vanadium can be seen in Fig. 3(b).

Figure 5 shows all the perfect matches of the [1$\bar{1}$0], [111], and [11$\bar{2}$] surfaces of both InAsSb and GaInAs. The red triangles in Fig. 5(a) therefore represents the same matches as those highlighted in Fig. 4. The surfaces have been chosen since they typically terminate nanowires of the two investigated alloys. Many zero strain solutions are possible, especially for the [111] surfaces. Furthermore, it is seen that the perfect matches are distributed well across the range of lattice constants which should make it easier to realize some of these interfaces experimentally. Details of all the matches of the remaining two surfaces of InAsSb and the three surfaces of GaInAs is in the Supplemental Material [32].

The method has also been used to find matches between the two semiconductors and the superconductor $Nb_{1-x}Ti_xN$. Since this material is an alloy in the NaCl structure, this is yet another case of varying the scaling parameter. We use a linear scaling of the NbTiN lattice constant in between the values

TABLE I. The perfect matches between NbTiN and the [1$\bar{1}$0], [111], and [11$\bar{2}$] surfaces of InAsSb and GaInAs. The NbTiN surface is the same as the given alloy surface.

InAsSb	k value	No. of cells	GaInAs	k value	No. of cells
[1$\bar{1}$0]	1.414	2	[1$\bar{1}$0]	1.354	11
[1$\bar{1}$0]	1.5	9	[111]	1.271	21
[111]	1.363	13	[111]	1.309	12
[111]	1.453	19	[111]	1.323	14
[111]	1.5	9	[111]	1.333	16
			[111]	1.363	13
			[11$\bar{2}$]	1.265	8
			[11$\bar{2}$]	1.291	5
			[11$\bar{2}$]	1.323	7

that can be found in the literature [28–31],

$$a_{\mathrm{NbTiN}}(y) = 4.30 + 0.17y \quad y \in [0;1]. \tag{20}$$

With this definition, the k value is given by $k = a_{\mathrm{alloy}}(x)/a_{\mathrm{NbTiN}}(y)$. The used matching parameters are the same as for the investigation of the pure metals and the perfect match results can be seen in Table I. For InAsSb, we find two zero strain matches for the [1$\bar{1}$0] surface and the first of these matches also has a very small unit cell of only two alloy surface unit cells. For the [111] surface, we find three zero strain matches and the $k = 1.5$ match has a reasonably small unit cell of nine surface cells. Plots like Figs. 4 and 5 showing all the matches between the semiconductors and NbTiN can be found in Supplemental Material [32].

VI. CONCLUSION

We have presented a general method for matching two crystals at an interface. A scaling parameter between the lattice constants of two crystals was introduced. Using this scaling parameter, we can consider a general match between an fcc and a bcc crystal. Matches for any value of the lattice constant ratio can be found by performing a single calculation. The method was applied to the two semiconductor alloys $InAs_{1-x}Sb_x$ and $Ga_xIn_{1-x}As$ matched with a range of metals (Al, Ni, Cu, Ag, Au, Pb, V, Fe, Nb, and Co). The scaling parameter was used to tune the lattice constant of the alloys with their composition. Results for the [1$\bar{1}$0], [111], and [11$\bar{2}$] alloy surfaces showed many perfect match solutions over a broad spectrum of the alloy lattice constant which is promising for realizing some of these matches experimentally. Finally, we have matched the two semiconductors to the superconducting alloy NbTiN. The results showed perfect matches with low surface cell areas for several of the investigated semiconductor surfaces.

ACKNOWLEDGMENTS

The authors acknowledges support from Innovation Fund Denmark under Project No. 5189-00082B "Atomic-scale modeling of interfaces" and the Quantum Innovation Center (QUBIZ), Innovation Fund Denmark, and from the European Commissions Seventh Framework Programme (FP7/20072013), Grant Agreement IIIV-MOS Project

No. 619326. We would like to thank Jesper Nygård and Thomas Sand Jespersen of the CQD at Copenhagen University for suggesting the idea of investigating semiconductor alloys, as well as Søren Smidstrup at QuantumWise A/S for the discussions and the help during the practical implementation of the matching method.

[1] A. Blom and K. Stokbro, J. Comput. Electron. **12**, 623 (2013).
[2] V. Dusastre, J. Heber, F. Pulizzi, A. Stoddart, P. Pamies, and C. Martin, Nat. Mater. **11**, 91 (2012).
[3] H. Kroemer, Rev. Mod. Phys. **73**, 783 (2001).
[4] Y. Yang, K. Wang, H.-W. Liang, G.-Q. Liu, M. Feng, L. Xu, J.-W. Liu, J.-L. Wang, and S.-H. Yu, Sci. Adv. **1**, e1500714 (2015).
[5] Y. Myung, D. M. Jang, T. K. Sung, Y. J. Sohn, G. B. Jung, Y. J. Cho, H. S. Kim, and J. Park, ACS Nano **4**, 3789 (2010).
[6] P. K. Kasanaboina, S. K. Ojha, S. U. Sami, C. L. Reynolds, Jr., Y. Liu, and S. Iyer, Semicond. Sci. Technol. **30**, 105036 (2015).
[7] A. L.-S. Chua, N. A. Benedek, L. Chen, M. W. Finnis, and A. P. Sutton, Nat. Mater. **9**, 418 (2010).
[8] D. Stradi, L. Jelver, S. Smidstrup, and K. Stokbro, J. Phys.: Condens. Matter **29**, 185901 (2017).
[9] P. Lazi, Comput. Phys. Commun. **197**, 324 (2015).
[10] P. Krogstrup, N. L. B. Ziino, W. Chang, S. M. Albrecht, M. H. Madsen, E. Johnson, J. Nygård, C. M. Marcus, and T. S. Jespersen, Nat. Mater. **14**, 400 (2015).
[11] W. Chang, S. M. Albrecht, T. S. Jespersen, F. Kuemmeth, P. Krogstrup, J. Nygård, and C. M. Marcus, Nat. Nanotechnol. **10**, 232 (2015).
[12] M. T. Deng, S. Vaitieknas, E. B. Hansen, J. Danon, M. Leijnse, K. Flensberg, J. Nygård, P. Krogstrup, and C. M. Marcus, Science **354**, 1557 (3016).
[13] A. P. Higginbotham, S. M. Albrecht, G. Kiranskas, W. Chang, F. Kuemmeth, P. Krogstrup, T. S. Jespersen, J. Nygård, K. Flensberg, and C. M. Marcus, Nat. Phys. **11**, 1017 (2015).
[14] S. M. Albrecht, A. P. Higginbotham, M. Madsen, F. Kuemmeth, T. S. Jespersen, J. Nygård, P. Krogstrup, and C. M. Marcus, Nature (London) **531**, 206 (2016).
[15] V. Mourik, K. Zuo, S. M. Frolov, S. R. Plissard, E. P. A. M. Bakkers, and L. P. Kouwenhoven, Science **336**, 1003 (2012).
[16] H. Zhang, Ö Gül, S. Conesa-Boj, K. Zuo, V. Mourik, F. K. de Vries, J. van Veen, D. J. van Woerkom, M. P. Nowak, M. Wimmer, D. Car, S. Plissard, E. P. A. M. Bakkers, M. Quintero-Pérez, S. Goswami, K. Watanabe, T. Taniguchi, and L. P. Kouwenhoven, arXiv:1603.04069.
[17] D. Stradi, S. Barja, C. Díaz, M. Garnica, B. Borca, J. J. Hinarejos, D. Sánchez-Portal, M. Alcamí, A. Arnau, A. L. Vázquez de Parga, R. Miranda, and F. Martín, Phys. Rev. B **88**, 245401 (2013).
[18] M. Batzill, Surf. Sci. Rep. **67**, 83 (2012).
[19] D. Stradi, S. Barja, C. Díaz, M. Garnica, B. Borca, J. J. Hinarejos, D. Sánchez-Portal, M. Alcamí, A. Arnau, A. L. Vázquez de Parga, R. Miranda, and F. Martín, Phys. Rev. Lett. **106**, 186102 (2011).
[20] K. Shoemake and T. Duff, *Proceedings of the Conference on Graphics Interface '92* (Morgan Kaufmann Publishers Inc., San Francisco, CA, 1992), pp. 258–264.
[21] We use the symmetry operations of the empty cell, since the algorithm does not take atomic positions into account.
[22] P. Niggli, *Krystallographische und Strukturtheoretische Grundbegriffe* (Akademische Verlagsgesellschaft, Leipzig, 1928).
[23] http://www.quantumwise.com; the VIRTUAL NANOLAB is freely available for academic researchers.
[24] The matching parameters $\ell_{\max}$, $A_{\max}$, $\varepsilon_{\max}$, and $m_{\max}$ are chosen as for the alloy investigations in Sec. V.
[25] I. P.-T. Institute, Electronic archive New semiconductor materials Characteristics and properties, http://www.ioffe.ru/SVA/NSM/Semicond/.
[26] C. Kittel, *Introduction to Solid State Physics*, 8th ed. (Wiley, New York, 2004).
[27] The area constraint is set for the cell size when using the lowest lattice constant of each alloy.
[28] K. Makise, H. Terai, M. Takeda, Y. Uzawa, and Z. Wang, IEEE Trans. Appl. Supercond. **21**, 139 (2011).
[29] P. Bosland, F. Guemas, M. Juillard, M. Couach, and A. Khoder, in *Proceedings of the Fifth Workshop on RF Superconductivity* (DESY, Hamburg, 1991), p. 497.
[30] T. Shiino, S. Shiba, N. Sakai, T. Yamakura, L. Jiang, Y. Uzawa, H. Maezawa, and S. Yamamoto, Supercond. Sci. Technol. **23**, 045004 (2010).
[31] H. Myoren, T. Shimizu, T. Iizuka, and S. Takada, IEEE Trans. Appl. Supercond. **11**, 3828 (2001).
[32] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevB.96.085306 for further results for interfaces between the semiconductor alloys InAsSb or GaInAs and metals or NbTiN.

Paper 3

Rich Ground State Chemical Ordering in a 309-Atom AgAu Icosahedral Nanoparticle
Larsen P M, Jacobsen K W and Schiøtz J

Preprint

Rich Ground State Chemical Ordering in a 309-Atom AgAu Icosahedral Nanoparticle.

Peter Mahler Larsen,* Karsten Wedel Jacobsen, and Jakob Schiøtz

Center for Atomic-scale Materials Design (CAMD), Department of Physics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

E-mail: pmla@fysik.dtu.dk

Abstract

In this article we determine the ground-state chemical ordering of AgAu 309-atom Mackay icosahedron nanoparticles. The energy of the nanoparticles is described using a cluster expansion model, and a Mixed Integer Programming (MIP) approach is used to find the ground state configurations for all stoichiometries. The chemical ordering varies widely between the different stoichiometries, and display a rich zoo of structures with non-trivial ordering.

Ever since the surprising discovery by Haruta *et al.* that gold is catalytically active in nanoparticulate form,[1] there has been intense research into the catalytic properties of gold[2–4] and silver[5] nanoparticles, including bimetallic Ag-Au nanoparticles.[6–8] These particles also display interesting optical and plasmonic properties, see e.g. the reviews by Feng *et al.*[9] and Boote *et al.*,[10] and show promising medical applications.[11]

The catalytic properties of a nanoparticle often depends critically on the detailed atomic configuration.[12,13] This is particularly important for bimetallic nanoparticles, which can preferentially exhibit one or the other material on the surface, and often can be designed in so-called core-shell structures with one of the metals as the catalytically active shell. It has been demonstrated that the ability to tailor materials that naturally forms desirable atomic-scale structures may significantly enhance catalytic activity and/or selectivity towards the desired reaction.[13]

It is thus important to be able to predict the shape and chemical ordering of nanoparticles.[14,15] This is, however, a difficult task both due to the difficulty of calculating the energy of a given configuration accurately, but mostly due to the very large configurational space that one must sample. This is usually done with Monte Carlo based techniques such as genetic algorithms,[16–18] simulated annealing,[19] basin hopping,[20] or minima hopping.[21] While all these methods can efficiently find configurations that are close to the global minimum, one cannot in principle know how close to the optimum the solutions are, nor can one know for sure if the global optimum has been found.

In this Letter we address the relatively simple case of bimetallic Au-Ag nanoparticles with 309 atoms in the Mackay icosahedral form. This is one of the so-called *magic number* structures where the per-atom energy is particularly low, leading to a morphology that is very robust to changes in stoichiometry and chemical ordering. We therefore only consider the chemical ordering, keeping the morphology constant. Even in this case, the search space is so large that it is unlikely that the stochastic methods find the ground state, for example there are $4.7 \cdot 10^{91}$ possible chemical orderings of the $Ag_{154}Au_{155}$ cluster. We address this by describing the energy of the nanoparticle using a Cluster Expansion model,[22] as this allows us to use a MIP[23,24]

approach to provably find the chemical ordering of the ground state configuration.

We find ground state configurations displaying a rich zoo of ordered structures, where the sites exposed on the nanoparticle surface vary dramatically with varying chemical composition of the nanoparticle. Figures 1(c) and 1(e) show nanoparticles with an onion-shell structure, and with flower-like vertex decoration. Neither structure is an intuitive ground state, fitting into none of the well-studied categories of core-shell, Janus, or phase mixing nanoparticles.[25]

Computational Approach

Given a fixed site geometry, a cluster expansion uses pseudo-spin variables at each site in conjunction with an orthogonal basis (the clusters) to model configurational properties of the system. More formally, a cluster hamiltonian is of the form:

$$\begin{aligned} E(\boldsymbol{\sigma}) =& V_0 + \sum_{i\in\mathbf{C}_1} V_i^{(1)}\sigma_i + \sum_{(i,j)\in\mathbf{C}_2} V_{i,j}^{(2)}\sigma_i\sigma_j \\ &+ \sum_{(i,j,k)\in\mathbf{C}_3} V_{(i,j,k)}^{(3)}\sigma_i\sigma_j\sigma_k + \dots \end{aligned} \tag{1}$$

where $E(\boldsymbol{\sigma})$ is the energy of a configuration $\boldsymbol{\sigma}$, $\mathbf{C}_n$ is the set of all n-body clusters, each containing cluster instances i, (i,j) or (i,j,k) for 1, 2 or 3-atom clusters respectively, collectively referred to as $\mathbf{c}_f$ in the following; $V_{c_f}^{(n)}$ are the effective cluster interactions (ECI) for the n-body cluster instances; and σ_i is the pseudo-spin variable at each site i.

A standard transformation is to change the spin variables $\sigma_i \in \{-1, 1\}$ to binary variables $x_i \in \{0, 1\}$ using the relation $\sigma_i = (2x_i - 1)$ which produces an equivalent Hamiltonian with different ECIs but only binary variables. It can be written compactly as

$$E(\boldsymbol{\sigma}) = E_0 + \sum_{\mathbf{c}_f} E_{\mathbf{c}_f} \prod_{i\in\mathbf{c}_f} \sigma_i, \tag{2}$$

where $E_{\mathbf{c}_f}$ are the new ECIs. With a Hamiltonian in this form, we can formulate a MIP model in order to find provably optimal configurations. MIP models, which are a generalization of linear programming models, solve problems of the form:

Minimize:	$\mathbf{c}^T\mathbf{x}$	Objective function
Subject to:	$\mathbf{A}\mathbf{x} \leq \mathbf{b}$	Constraints

A linear program consists of a set of n continuous variables $\mathbf{x} \in \mathbb{R}^n$, an associated set of costs for each variable $\mathbf{c} \in \mathbb{R}^n$, and a set of m linear constraints, denoted here by $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$. The goal, or *objective function*, is to find values for the set $\mathbf{x}$ such that the total cost is provably minimized, whilst respecting the constraints. In a MIP model, some or all of the variables are furthermore constrained to have integer values.

Model 1 shows the MIP model for determining the ground state chemical ordering of a bimetallic nanoparticle. Each predetermined site, with index i, has an associated binary variable x_i (3) which determines whether an A-type or B-type atom is placed at that site. The system contains N_B B-type atoms, which is enforced by equation (8). The activity of a cluster instance, indicated by a binary variable $y_{\mathbf{c}_f}$ (4), is governed by equations (9) and (10); taken together, these constraints are equivalent to the relation $y_{\mathbf{c}_f} = \prod_{i\in\mathbf{c}_f} x_i$. Lastly, associated with each cluster is a predetermined ECI (5) which is used to determine the total energy of the system (7). Thus, the objective of the model is to choose how to order the A-type and B-type atoms such that the total energy of the system is minimized.

A different approach to finding provably optimal ground states was recently demonstrated by Huang et al.,[26] making use of pseudo-Boolean optimization rather than MIP, and applying it to bulk alloys.

Cluster Selection and ECI Fitting

For a system as large as a 309-atom nanoparticle, the ab initio energy calculation of even a single configuration can take days. As such, ab initio calculations cannot be used to sample a sufficient number of configurations for a CE

Variables:	$x_i \in \{0,1\}$	$\forall i$	Type of atom site i (A=0, B=1)	(3)
	$y_{\mathbf{c}_f} \in \{0,1\}$	$\forall \mathbf{c}_f$	Cluster instance variable (off=0, on=1)	(4)
Parameters:	$E_{\mathbf{C}} \in \mathbb{R}$	$\forall \mathbf{C}$	Energy of cluster $\mathbf{C}$	(5)
	$N_B \in \mathbb{N}$		Number of B-type atoms	(6)
Minimize:	$\sum_{\mathbf{C}} \sum_{\mathbf{c}_f \in \mathbf{C}} E_{\mathbf{C}} y_{\mathbf{c}_f}$		Total energy of system	(7)
Subject to:	$\sum_i x_i = N_B$		Fixed number of B-type atoms	(8)
	$y_{\mathbf{c}_f} \leq x_i$	$\forall i \in \mathbf{c}_f$	Any atom absent from $\mathbf{c}_f \Rightarrow y_{\mathbf{c}_f}$ off	(9)
	$y_{\mathbf{c}_f} \geq 1 - \lvert\mathbf{c}_f\rvert + \sum_{i \in \mathbf{c}_f} x_i$		All atoms present in $\mathbf{c}_f \Rightarrow y_{\mathbf{c}_f}$ on	(10)

Model 1: A MIP model for determining the configuration which provably minimizes the energy of a CE model. The placement of the sites as well as the ECIs are fixed; the model determines the optimal chemical ordering only. For any cluster instance with a positive [negative] ECI, the constraint given by equation (9) [equation (10)] is redundant.

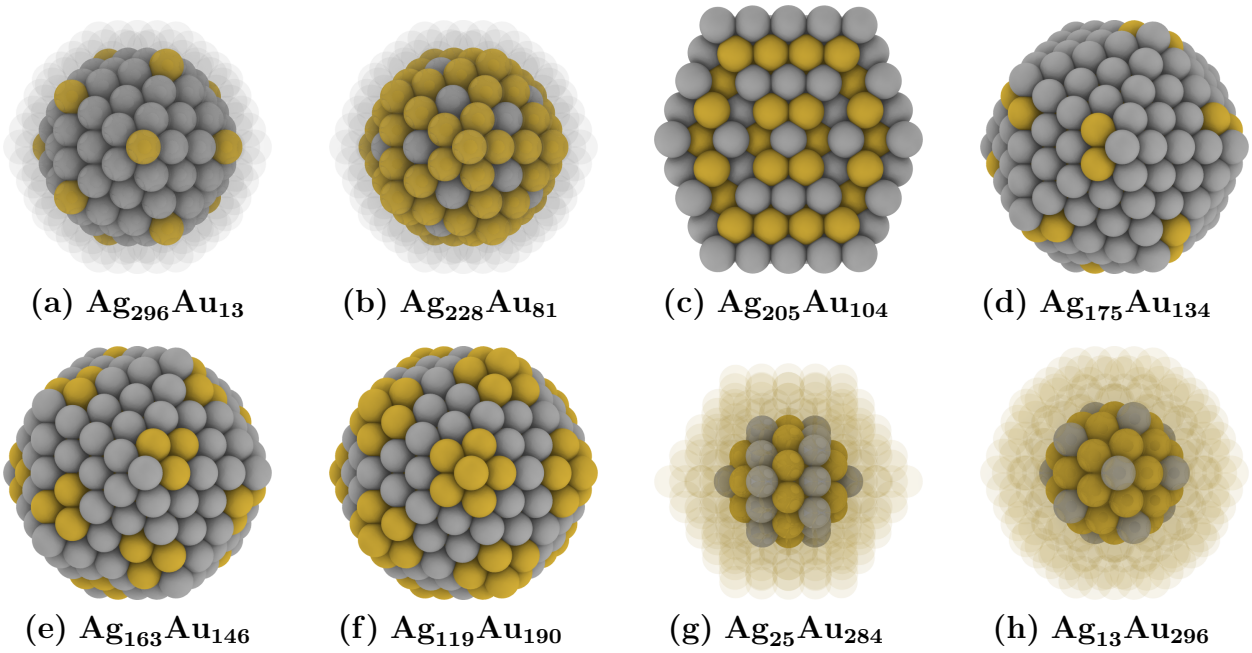
model in a reasonable timescale. Instead, we use the semi-empirical Effective Medium Theory (EMT)[27] potential to calculate energies. Semi-empirical potentials are fast to evaluate and highly accurate in bulk systems, but typically mispredict the energies of surface atoms. Nonetheless, whilst the energies might be quantitively inaccurate, the different site types in a 309-atom nanoparticle have such great variation in energy that we can expect the energy difference of different configurations to be at least qualitatively correct.

The current best practice for cluster and ECI selection is to use compressive sensing,[28,29] though this is motivated by the need to construct a sparse model from limited data. The use of a semi-empirical potential, however, permits the sampling of a vast number of configurations, which in turn allows for a simple calculation of the ECIs. We have sampled 45,000 chemical ordering configurations, using a mixture of random sampling and stochastic local search in order to include configurations with both high and low energies (c.f. Figure 2); sampling a wider range of energies in this way has been shown to improve the generalization of a CE model.[30] The energy of each configuration has been minimized using gradient descent to allow local relaxations without changing the overall structure of the nanoparticle. To select clusters, we generate all 1, 2, and 3-body clusters with a maximum site-site distance of three, where the inter-site distance is given by the number of nearest-neighbour 'hops' between the sites. From the resulting 2699 clusters, we identify relevant clusters with iterated stochastic local search, producing a smaller CE model with 55 1-body, 2-body and 3-body clusters, which contain a total of 3795 cluster instances. The resulting CE model is robust; when measured against a set of $20,000$ configurations not used in the fitting process, the CE model has a root-mean-square error (RMSE) of 0.078 meV/atom, which increases to only 0.166 meV/atom when measured against the ground-state structures found using the MIP model.

It should be noted that due to the finite range of a cluster expansion, it sometimes will have a degenerate ground state; a degeneracy lifted in the original potential by weak long-range elastic interactions.

Results and Discussion

We have used the CE model described to find the optimal configuration of the nanoparticle at every composition between 0-309 Au atoms.



(a) $Ag_{296}Au_{13}$ (b) $Ag_{228}Au_{81}$ (c) $Ag_{205}Au_{104}$ (d) $Ag_{175}Au_{134}$

(e) $Ag_{163}Au_{146}$ (f) $Ag_{119}Au_{190}$ (g) $Ag_{25}Au_{284}$ (h) $Ag_{13}Au_{296}$

Figure 1: A selection of ground-state configurations in a 309-atom Mackay icosahedron AgAu nanoparticle. The configurations exhibit a diverse range of highly ordered interior and surface structures. In **(a)** and **(b)**, the transparent atoms represent a single layer of Ag atoms; in **(g)** and **(h)** they represent two layers of Au atoms. The Au atoms exhibit a strong preference for subsurface shell sites, in particular the subsurface corners **(a)**, followed by the subsurface edges **(b)**. The Ag atoms show a strong preference for 2nd shell corner sites **(g)**-**(h)**, as well as the central atom. The lowest energy configuration **(c)** (shown as a slice through the nanoparticle) has a perfectly ordered onion-shell structure. At high Au concentrations, the preference of Au atoms for corner, edge or interior sites of the surface facets varies as a function of composition **(d)**-**(f)**.

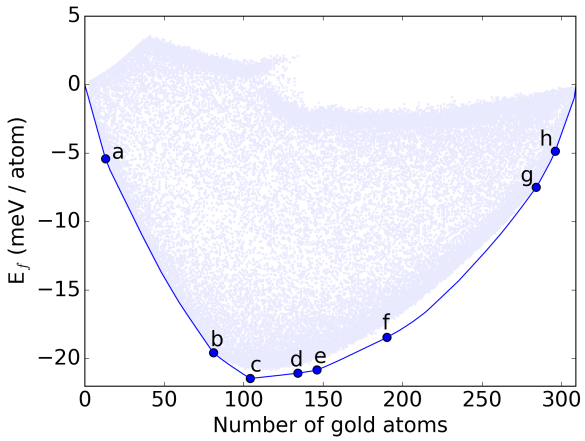


Figure 2: Energy of formation (from EMT) vs. the number of Au atoms in the nanoparticle. The grey dots show configurations sampled for CE model construction. The blue line shows the convex hull of the ground state configurations. Compositions at which the nanoparticle exhibits strong ordering (shown in Figure 1) are marked with a circle.

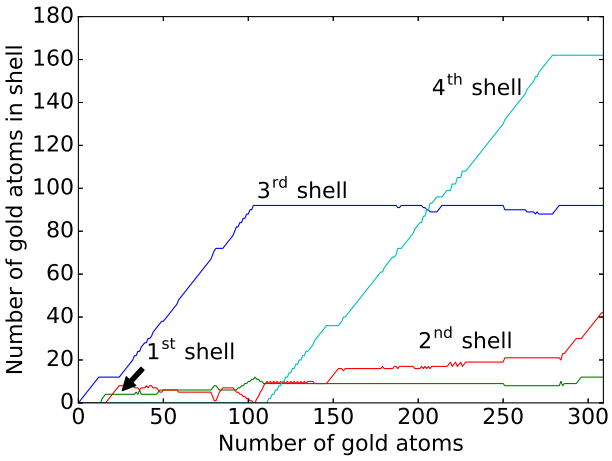


Figure 3: Number of Au atoms in each shell vs. the total number of Au atoms in the nanoparticle. The structural evolution exhibits a complex interplay between the shells, with no monotonic components.

The nanoparticle exhibits a surprisingly complex structural evolution. Figure 3 shows the number of Au atoms in each shell as a function of composition. Quite noticable is that no shell shows a monotonic increase in Au content. Here, the second shell is particularly illustrative, in that it increases in Au content from 42 Au atoms, but has zero Au content again at 104 Au atoms. A further example (not shown in Figure 3) is the central atom, which is occupied by a Au atom at 13 and 309 Au atoms only.

Figure 2 shows the convex hull of the formation energy of the ground-state configurations at every composition. The formation energy of a configuration $\vec{\sigma}$ with fractional compositions x_{Ag} and x_{Au} is given by:

$$E_f(\vec{\sigma}) = \frac{1}{309}\left[E(\vec{\sigma}) - x_{\mathrm{Ag}}E_{\mathrm{Ag}} - x_{\mathrm{Au}}E_{\mathrm{Au}}\right] \quad (11)$$

The convex hull contains a high number of compositions (83), which causes it to be a largely smooth function of composition. As such, the compositions highlighted are those where the energy gradient changes rapidly. These configurations (shown in Figure 1) are remarkable in that they exhibit strong ordering, either rotationally symmetric ordering (Figures 1(a)-(c), (f)- (h)), or ordered geometric patterns (Figures 1(d), (e)).

All nanoparticles with up to 111 Au atoms only present Ag atoms in the surface, and all particles with more than 279 Au atoms only present Au atoms in the surface, in spite of the lower surface energy of Ag. In many cases, regular patterns are formed inside the nanoparticle. Of particular interest is the $Ag_{205}Au_{104}$ cluster shown in Figure 1(c), forming an onion-like structure with alternating layers of pure Ag and Au. This is also the cluster displaying the most negative formation energy.

The nanoparticles in Figure 1(d) and (e) show how Au atoms tend to form two or three atom islands next to, but not at the corners of the icosahedron. When the Au content is increased, however, the Au forms larger islands which switch to being centered on the corners.

Overall, the structural evolution as a function of composition is driven by a trade-off between the preference for Ag and Au to form ordered mixtures (as they do in bulk materials at 0K) and the large energetic differences between different site types. We note that, regardless of the energetic model used, Monte Carlo methods are incapable of determining the optimality of a configuration; the conclusions of the present work rely on the ability of the MIP model to guarantee that the structures found are indeed the ground state structures.

Acknowledgements

P.M.L. thanks J. Connelly for productive discussions, C. Schuh for an introduction to the nanoparticle design problem, and J. Larsen for advice on selecting solver settings. This work was supported by a research grant (9455) from VILLUM FONDEN, and by grant 1335-00027B from the Danish Council for Independent Research.

References

(1) Haruta, M.; Kobayashi, T.; Sano, H.; Yamada, N. *Chemistry Letters* **1987**, *16*, 405–408.

(2) Daniel, M.-C.; Astruc, D. *Chemical Reviews* **2004**, *104*, 293–346.

(3) Ishida, T.; Koga, H.; Okumura, M.; Haruta, M. *Chem Rec* **2016**, *16*, 2278–2293.

(4) Sardar, R.; Funston, A. M.; Mulvaney, P.; Murray, R. W. *Langmuir* **2009**, *25*, 13840–13851.

(5) Guo, J.-Z.; Cui, H.; Zhou, W.; Wang, W. *Journal of Photochemistry and Photobiology A: Chemistry* **2008**, *193*, 89–96.

(6) Liu, J.-H.; Wang, A.-Q.; Chi, Y.-S.; Lin, H.-P.; Mou, C.-Y. *J. Phys. Chem. B* **2005**, *109*, 40–43.

(7) Chang, C. M.; Cheng, C.; Wei, C. M. *J. Chem. Phys.* **2008**, *128*, 124710–124710.

(8) Raveendran, P.; Fu, J.; Wallen, S. L. *Green Chem.* **2006**,

(9) Feng, L.; Gao, G.; Huang, P.; Wang, K.; Wang, X.; Luo, T.; Zhang, C. *Nano BioMed ENG* **2010**, *2*, 1–10.

(10) Boote, B. W.; Byun, H.; Kim, J.-H. *Journal of Nanoscience and Nanotechnology* **2014**, *14*, 1563–1577.

(11) Bachelet, M. *Mater. Sci. Techn.* **2016**, *32*, 794–804.

(12) Honkala, K.; Hellman, A.; Remediakis, I. N.; Logadottir, A.; Carlsson, A.; Dahl, S.; Christensen, C. H.; Norskov, J. K. *Science* **2005**, *307*, 555–558.

(13) Siahrostami, S.; Verdaguer-Casadevall, A.; Karamad, M.; Deiana, D.; Malacrida, P.; Wickman, B.; Escudero-Escribano, M.; Paoli, E. A.; Frydendal, R.; Hansen, T. W.; Chorkendorff, I.; Stephens, I. E. L. S.; Rossmeisl, J. *Nat. Mater.* **2013**, *12*, 1137–1143.

(14) Ferrando, R.; Jellinek, J.; Johnston, R. L. *Chemical Reviews* **2008**, *108*, 845–910.

(15) Rossi, G.; Rapallo, A.; Mottet, C.; Fortunelli, A.; Baletto, F.; Ferrando, R. *Phys. Rev. Lett.* **2004**, *93*, 105503.

(16) Lysgaard, S.; Mýrdal, J. S. G.; Hansen, H. A.; Vegge, T. *Phys Chem Chem Phys* **2015**, *17*, 28270–28276.

(17) Hartke, B. *The Journal of Physical Chemistry* **1993**, *97*, 9973–9976.

(18) Rossi, G.; Ferrando, R.; Rapallo, A.; Fortunelli, A.; Curley, B. C.; Lloyd, L. D.; Johnston, R. L. *J. Chem. Phys.* **2005**, *122*, 194309–194309.

(19) Lee, J.; Lee, I.-H.; Lee, J. *Phys. Rev. Lett.* **2003**, *91*, 080201–4.

(20) Wales, D. J.; Scheraga, H. A. *Science* **1999**, *285*, 1368–1372.

(21) Goedecker, S. *J. Chem. Phys.* **2004**, *120*, 9911–9917.

(22) Sanchez, J.; Ducastelle, F.; Gratias, D. *Physica A* **1984**, *128*, 334–350.

(23) Nemhauser, G. L.; Wolsey, L. A. *Integer Programming and Combinatorial Optimization*; John Wiley & Sons, 1988.

(24) Chen, D.-S.; Batson, R. G.; Dang, Y. *Applied integer programming: modeling and solution*; John Wiley & Sons, 2010.

(25) Wang, H.; Chen, L.; Feng, Y.; Chen, H. *Acc. Chem. Res.* **2013**, *46*, 1636–1646.

(26) Huang, W.; Kitchaev, D. A.; Dacek, S. T.; Rong, Z.; Urban, A.; Cao, S.; Luo, C.; Ceder, G. *Phys. Rev. B* **2016**, *94*.

(27) Jacobsen, K.; Stoltze, P.; Nørskov, J. *Surface science* **1996**, *366*, 394–402.

(28) Nelson, L. J.; Hart, G. L. W.; Zhou, F.; Ozolins, V. *Phys. Rev. B* **2013**, *87*.

(29) Nelson, L. J.; Ozolins, V.; Reese, C. S.; Zhou, F.; Hart, G. L. W. *Phys. Rev. B* **2013**, *88*.

(30) Seko, A.; Koyama, Y.; Tanaka, I. *Phys. Rev. B* **2009**, *80*.

Paper 4

Mixed Integer Programming for Nanoparticle Design
Larsen P M, Jacobsen K W and Schiøtz J
In preparation
Submission pending publication of Paper 3.

Mixed Integer Programming for Nanoparticle Design

Peter Mahler Larsen, Karsten Wedel Jacobsen, Jakob Schiøtz

Center for Atomic-scale Materials Design, Department of Physics, Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark

1. Introduction

Metallic nanoparticles play a central role in heterogeneous catalysis, which in turn is used everywhere in modern society from the chemical industry to enviromental applications e.g. in cars. For this reason, the shape and functionality of metallic nanoparticles has long been a very active research field.

As many catalytic reactions are strongly dependent on the detailed atomic configuration, details in the atomic-scale structure of the nanoparticles may play a central role for the performance as a catalyst. For example, the active sites for ammonia synthesis (a process accounting for 1% of the world's energy consumption [1]) are five-fold coordinated sites at the foot of atomic-scale steps on the surface [2], and similar sites on nanoparticles [3]. For other reactions, undercoordinated atoms at edges [XXX] or corners [4, 5] are the active part of the nanoparticle. For alloy nanoparticles containing two or more metals, the chemical ordering and surface segregation also becomes important.

The traditional approach for finding ground state nanoparticle configurations is to use ab-initio energy calculations or semi-empirical potentials in combination with stochastic search methods, such as Metropolis Monte Carlo on a lattice [XXX], multi-scale Monte Carlo methods [6, 7], genetic algorithms [8, 9], simulated annealing [10], basin hopping [11] or minima hopping [12] (c.f. Rossi et al. [13] for a good overview). Whilst the energetic calculations are at least *reasonably* accurate, no stochastic search method is guaranteed to find the minimum energy configuration. Furthermore, if the ground state configuration is found, there is no way of *knowing* it has been found.

Our approach is to turn the problem on its head; instead of using an exact potential and searching for an approximate minimizer of the energy, we use an approximate potential and find the *provably optimal* ground state configuration. We find the ground states using a Mixed Integer Programming (MIP) model. MIP is widely used in management sciences for solving problems as diverse as nurse roster scheduling [14, 15], shipping container stowage [16], airline crew scheduling [17, 18], capacitated vehicle routing [19, 20], the cutting-stock problem [21, 22], and the well-known travelling salesman problem [23]. In the physical sciences MIP modelling has seen little use,

though some notable exceptions include the determination of ground states in Ising spin glasses [24, 25], radiation therapy beam angle selection [26], and local structural alignment in proteins [27]. Recently, Huang et al. [28] have developed a method using pseudo-Boolean optimization to find the ground state configurations of generalized Ising models.

In this paper, we briefly present the ideas behind MIP in section 2. We then present an algorithm for finding the structure of a nanoparticle consisting of a single chemical element on a regular crystalline lattice in section 3. In sections 4 and 5 we consider chemical ordering in a nanoparticle with fixed morphology.

2. Mixed Integer Programming

A detailed description of MIP modelling can be found in the standard texts [29, 30], but since it forms the core of this article and since few in the physics community are familiar with it, we will briefly describe its concepts here.

MIP modelling is a generalization of linear programming (LP) which is used to solve problems of the form:

Minimize:	$\mathbf{c}^T\mathbf{x}$	Objective function
Subject to:	$\mathbf{A}\mathbf{x} \leq \mathbf{b}$	Constraints

A linear program consists of a set of n continuous variables $\mathbf{x} \in \mathbb{R}^n$, an associated set of costs for each variable $\mathbf{c} \in \mathbb{R}^n$, and a set of m linear constraints, denoted here by $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. The goal, or *objective function*, is to find values for the set $\mathbf{x}$ such that the total cost is minimized, whilst respecting the constraints. The constraints describe the space of valid solutions; since all the constraints are linear, the solution space is a *convex polytope*. Since the objective function is a linear function of $\mathbf{x}$, the optimal solution lies at a vertex of the polytope. These concepts are illustrated for a simple two-variable model in Figure 1.

A MIP model is a generalization of a linear programming model, whereby some or all of the variables are constrained to have integer values. Integrality constraints are fundamentally non-linear, since the integers do not form a continuous set. As a consequence, most MIP models are considerably harder to solve than their linear counterparts. The main difficulty is highlighted in Figure 2; the optimal integral solution does not necessarily lie at a vertex of the convex polytope, and can in fact lie very far from a vertex.

Whilst the number of solution methods are vast, most modern MIP solvers use a variation of a *branch and bound* algorithm to find the optimal integral solution. In a branch and bound process, the set of all possible solutions is represented as a tree; each level of the tree corresponds to a variable, and the branches at each level correspond to the value assigned to the variable. Thus, the leaves of the tree represent all possible solutions. The optimal solution is found by exploring the branches of the tree. Exploring all the branches is no quicker than an exhaustive search of all possible solutions, but the

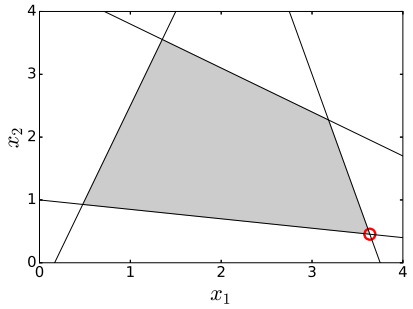


Figure 1: A two-dimensional LP model. The objective is to minimize $x_2 - x1$ subject to the constraints $4x_1 + x_2 \leq 15$, $-0.15x_1 - x_2 \leq -1$, $0.7x_1 + x_2 \leq 4.5$, $-3x_1 + x_2 \leq -0.5$. The optimal solution lies at $3.64, 0.45$), shown by the red circle. The feasible region is shaded grey.

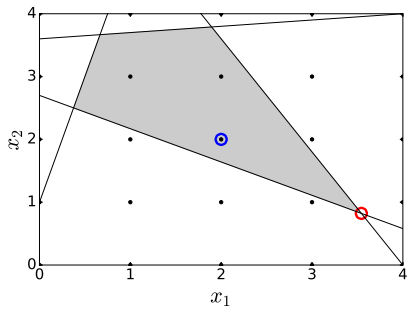


Figure 2: A two-dimensional MIP model. The optimal integer solution (blue circle) lies far from the optimal linear relaxation of the model (red circle). Rounding the fractional solution does not produce a solution which lies in the feasible region.

search tree can be efficiently pruned using a lower bound; given an incumbent integral solution, an entire branch can be discarded, or *pruned*, if the lower bound of the objective function is higher than that of the cost of the incumbent. In a MIP model, a lower bound can be obtained by solving the model without integrality constraints on the variables at the lower levels of the search tree. Solving a model without integrality constraints is known as the *linear relaxation*.

Off-the-shell commercial MIP solvers use the branch and bound algorithm, as well as many other ingenious mathematical tricks, to quickly solve models containing thousands or even hundreds of thousands of integer variables. Even when the model cannot be solved to optimality, the linear relaxation establishes a lower bound on the value of the optimal solution. In many cases, knowing that the cost incumbent solution is e.g. at most 0.6% from that of the optimal solution is useful knowledge in itself, and knowledge which is not available when using stochastic search methods.

3. Single-Element Nanoparticle Morphology

In the previous section we described MIP models. Here, we describe how MIP models can be applied to the problem of finding the ground-state morphology of a single-element nanoparticle. At small nanoparticle sizes, morphology changes rapidly as a function of the number of atoms in the system. For example, Baletto et al. [31] have shown that Pt nanoparticles with N atoms show a preference for icosahedral structure up to $N \approx 80$, decahedral structure in the range $\approx 100 \leq N \leq \approx 1500$, and truncated octahedral structure (FCC) for $N \geq \approx 1500$. The size dependency varies significantly depending on the element, and in many cases the energy difference between morphologies is very small at small values of N. The model we have developed requires a fixed set of possible atomic sites. This model is therefore most appropriate at larger sizes

Indices:	$i, j \in \mathbf{I}$		Site index	(1)
Variables:	$x_i \in \{0, 1\}$	$\forall i$	Atom presence at site i	(2)
	$y_{ij} \in \{0, 1\}$	$\forall i, j$	Bond presence between sites i and j	(3)
Parameters:	$E_{ij} \in \mathbb{R}$	$\forall i, j$	Energy of bond between sites i and j	(4)
Minimize:	$\sum_{ij} E_{ij} y_{ij}$		Total energy of system	(5)
Subject to:	$\sum_{i} x_i = N$		Fixed number of atoms	(6)
	$y_{ij} \leq x_i$	$\forall i, j$	Atom absent from site $i \Rightarrow$ bond absent	(7)
	$y_{ij} \leq x_j$	$\forall i, j$	Atom absent from site $j \Rightarrow$ bond absent	(8)
	$y_{ij} \geq x_i + x_j - 1$	$\forall i, j$	Atoms at sites i and $j \Rightarrow$ bond present	(9)

Model 1: A MIP model for minimizing the energy of a single-element nanoparticle. The placement of the sites as well as the bond energies are fixed; the model decides at which sites to place atoms. The energetic model consists of pairwise bonds only. If all bond energies are negative, the constraint given by equation (9) is redundant. Solution time can be reduced with symmetry-breaking constraints; this is discussed in section 5.

where the nanoparticle is known to be FCC or BCC. Furthermore, the potential we use is inappropriate for very small systems. For very small nanoparticles only ab initio energy calculations are suitable. These cannot be formulated as MIP models. At larger length scales semi-empirical interatomic potentials [32, 33, 34] can be used. Semi-empirical potentials typically consist of a pairwise bonding term, and an electron density dependent embedding function. In section 4 we show how these can be incorporated into a MIP model, but here we use an energy expression which includes only a pairwise bonding term.

Model 1 shows the MIP model for determining the ground state morphology of a single-element nanoparticle. Each predetermined site, with index i (1), has an associated binary variable x_i (2) which determines whether or not an atom is placed at that site. The system contains N atoms, which is enforced by equation (6). A bond can exist between each pair of sites, which is indicated by a binary variable y_{ij} (3). A bond can *only* exist if atoms are present at both sites i and j (7)-(8) and *must* exist if both atoms are present (9); these equations constitute a *logical AND* constraint [35] where $y_{ij} = x_i$ AND x_j. Lastly, associated with each bond is a predetermined energy (4) which is used to determine the total energy of the system (5). Thus, the objective of the model is to choose where to place the atoms such that the total energy of the system is minimized. The remaining choices to make are where to place the the sites and which

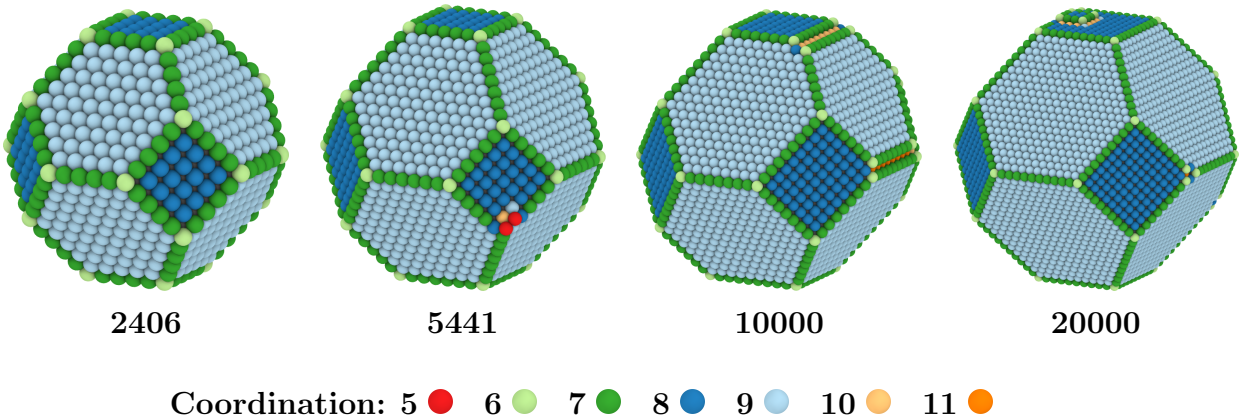


Figure 3: Ground state configurations for a parameter-free energetic model with nearest-neighbour bonding only. The 2406-atoms nanoparticle is a 'magic' truncated octahedron, which exhibits cubic symmetry. The other structures exhibit defects, though relatively few. The 11-coordinated atoms in the 10000 atom system do not constitute true 110 facets but single-row defects; given more atoms the defects will be filled in.

energies to assign to the bonds. We have placed the sites in a FCC lattice structure. Solution time scales typically scales exponentially with the number of variables, so we have 'frozen' an inner core of atoms, and placed a few layers of non-frozen sites around the core. Care should be taken that the structure of the frozen core and the non-frozen sites does not prevent discovery of the optimum configuration. The simplest bond energy choice is to set $E_{ij} = -1$ if i and j are nearest neighbours, and $E_{ij} = 0$ otherwise. This is essentially a parameter-free model and is equivalent to maximizing the number of nearest neighbour bonds. Using this energetic model we have solved every structure for up to 5000 atoms for both FCC and BCC nanoparticle structures, which are available for download in the Computational Materials Repository [36].

Figure 3 shows the ground state configurations for a selection of nanoparticle sizes. A few features are immediately obvious. The structures have few defects, and at most one 'island' on top of a surface. There are no 110 facets present; indeed, the Wulff construction for this energetic model contains no 110 facets (c.f. Figure 7). Furthermore, the structures exhibit a degree of scale invariance; whilst the relative numbers of edges, terraces and interior atoms change, the overall structure remains that of a truncated octahedron. In their seminal paper on surface site statistics [37] Van Hardefeld and Hartog studied 'magic' truncated octahedra, which exhibit perfect cubic symmetry; the eight 111 facets are identical, the six 100 facets are identical, and all edges (including corners) separating the facets contain m atoms. Magic truncated octahedra occur at:

$$N = 16m^3 - 33m^2 + 24m - 6 \tag{10}$$

where m is an integer. The number of nearest-neighbour bonds for a magic truncated

octahedron is given by:

$$f(m) = 192m^3 - 492m^2 + 420m - 120 \tag{11}$$

Our structures fill in the gaps between the magic numbers. Figure 4 shows the difference between the number of nearest-neighbour bonds in a ground-state structure and a magic truncated octahedron, as a function of nanoparticle size. Since magic structures only occur for integer values of m, the difference is for hypothetical magic structures with non-integral values of m. It can be seen that magic truncated octahedra correspond to a local maximum in the number of bonds. Indeed, local maxima occur each time a line or surface is completed. The dip after surface completion is due to the presence of an atom with a coordination number of four; there is nowhere else to place it but on top of a surface. The number of bonds is maximized when this atom is placed on a 100 facet. Once the nanoparticle is large enough that it has settled into a stable truncated octahedron, at most one four-coordinated atom can be present in any structure, since the addition of another atom produces two five-coordinated atoms. For the same reason, at most two five-coordinated atoms can be present. Figure 5 shows the coordination counts as a function of nanoparticle size. As expected, the number of edge atoms (7-coordinated) scale proportionally to $N^{\frac{1}{3}}$, the surface atoms (8- and 9-coordinated) proportionally to $N^{\frac{2}{3}}$, and the bulk atoms (12-coordinated) proportionally to N. The number of corner atoms hovers around 24, which is the number of corners in a magic truncated octahedron. In some structures there are only 23 6-coordinated atoms. This can be seen for the 5441-atom nanoparticle in Figure 3; a surface atom is placed next to what was previously a corner atom, thereby increasing its coordination. This highlights the inherent degeneracy in the chosen energetic model; there exist multiple structures with the same number of nearest-neighbour bonds, but with different coordination numbers.

Whilst the parameter-free energetic model is a good starting point for investigating nanoparticle morphology at different sizes, it is not a good model for all chemical elements. By extending the interaction distance between atoms we can represent different potentials and thereby change the obtained nanoparticle morphology. Figure 6 shows the energy per atom as a function of coordination using the EMT potential for Ni. The EMT potential has a cutoff between the third and fourth neighbour shells, so in a perfect lattice there are exactly three bond types; 12 nearest-neighbour (first-shell) bonds, 6 next-nearest-neighbour (second-shell) bonds, and 24 third-shell bonds. In total there are therefore 1728 possible per-atom energies, given that relaxations are forbidden. It is quite remarkable that, for some elements at least, a linear fit of the three bond types produces a negligible deviation from the exact EMT potential for coordinations with at least 6 nearest neighbours. For lesser coordinations the linear model underestimates the energy, but this is of little practical concern since 4- and 5-coordinated atoms occur very infrequently, and lesser coordinated atoms do not occur in all but the very smallest nanoparticles. In order to optimize nanoparticle morphologies based on EMT, we have used a linear fit using only the first and second shell neighbour bonds. The third shell neighbour bonds make only a small contribution to the energy at

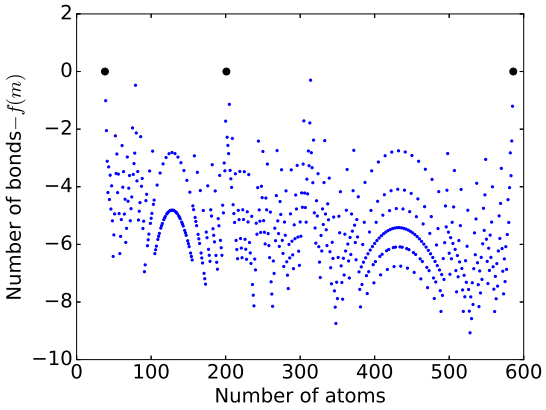


Figure 4: The difference in the number of bonds in the ground state configuration and a hypothetical magic truncated octahedron with the same number of atoms, for all N between the first three magic structures. The term $f(m)$ is defined in equation (11). The black dots denote actual magic truncated octahedra.

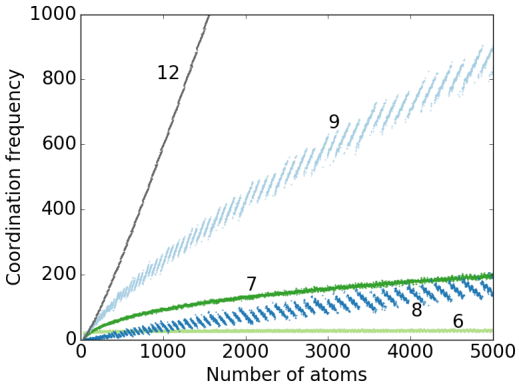


Figure 5: Coordination counts for ground state configurations with nearest-neighbour only bonds, for structures with up to $N = 5000$ atoms.

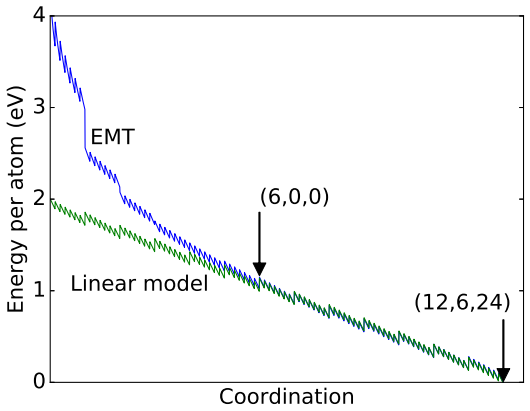


Figure 6: Energy per atom as a function of increasing coordination, for both the EMT potential for Ni, and an approximation using a linear function of the coordination numbers. The x-axis is the lexicographically sorted coordination vector, using the counts of the first three neighbour shells. Coordination vectors less than $(6, 0, 0)$ are not included in the fit. The maximum absolute error in the range $(6, 0, 0) - (12, 6, 24)$ is 0.011eV/atom. This increases to 0.065eV/atom when only first and second shell neighbours are used.

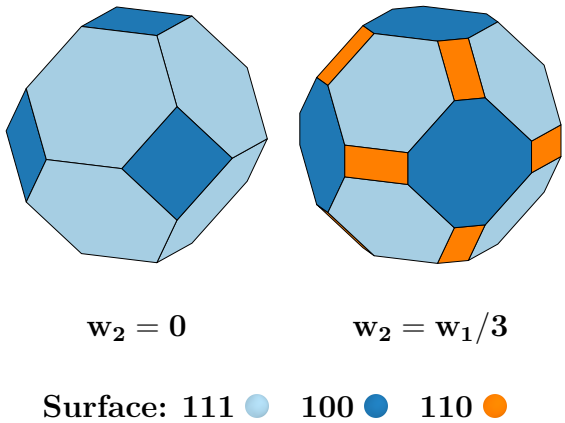


Figure 7: Wulff constructions for two different relative weightings of the next-nearest-neighbour to the nearest-neighbour bond energies. Third neighbour shell bonds are not included. In a model with nearest-neighbour bonds only $(w_1 = 1, w_2 = 0)$ the 110 surfaces are not present. Increasing the relative weighting increases the proportion of the 110 and 100 surfaces at the expense of the 111 surfaces.

the expense of increasing the number of variables in the MIP model by more than a factor of two. Including these bonds would significantly reduce the size of the nanoparticles we can solve. Nonetheless, even with two neighbour shell bond types we can create nanoparticles with very different morphologies.

Figure 7 shows the Wulff constructions for two different weightings of the next-nearest-neighbour bonds relative to the nearest-neighbour bonds. It can be seen that the 110 surface is not present in the model with nearest-neighbour bonds only. As the relative contribution of the next-nearest-neighbours increases, so do the relative proportions of the 110 and 100 surfaces. Using the linear model shown in Figure 6, we have found configurations for nanoparticles with 10000 and 20000 atoms. Figure 8 shows both the whole nanoparticles and a slice through the 110 direction with the Wulff constructions superimposed. At both sizes the nanoparticles have no surface islands, and relatively few defects. Interestingly, the defects accumulate around the 110 surfaces. It can also be seen that, at these sizes, the difference between the discrete structures and the continuous Wulff constructions is very small.

It can be argued that a linear model is too great a simplification of a semi-empirical potential with a complex functional form and many parameters. However, a semi-empirical potential must capture many physical properties, such as lattice structure, bulk modulus, surface energy and vacancy formation energy. In our model, the use of predetermined sites imposes the correct lattice structure on the resulting nanoparticle. Thereafter, the only properties of interest are the *relative* surface energies. As such, a single-parameter model is quite adequate. The structures we have obtained are, even visibly, very low energy configurations, which would be exceedingly difficult to obtain using more traditional stochastic search methods.

4. Optimal Chemical Ordering using Cluster Expansions

In the previous section we showed how the ground state morphology can be found for a single-element nanoparticle. A similar method can be applied to find the optimal chemical ordering in a two-element nanoparticle. Rather than determining the placement of atoms, the problem is that of deciding which of two elements to place at each site, whilst conserving the stoichiometry. Thus, the morphology is fixed but the *permutation* of the elements is variable. There exists a wide array of methods for finding ground state configurations, both with atomic positions as variables and with fixed sites, but here, we focus on the *cluster expansion* method [38] which has seen many successful nanoparticle applications [39, 40, 41, 42, 43, 44].

In addition to the fixed site geometry, a cluster expansion uses pseudo-spin variables at each site in conjunction with an orthogonal basis (the clusters) to model configurational properties of the system. More formally, a cluster hamiltonian is of the form:

$$E(\sigma) = V_0 + \sum_{\mathbf{c}_f \in \mathbf{C}_1} \sum_{i \in \mathbf{c}_f} V_{1,f}\sigma_i + \sum_{\mathbf{c}_f \in \mathbf{C}_2} \sum_{(i,j) \in \mathbf{c}_f} V_{2,f}\sigma_i\sigma_j + \sum_{\mathbf{c}_f \in \mathbf{C}_3} \sum_{(i,j,k) \in \mathbf{c}_f} V_{3,f}\sigma_i\sigma_j\sigma_k + \dots \quad (12)$$

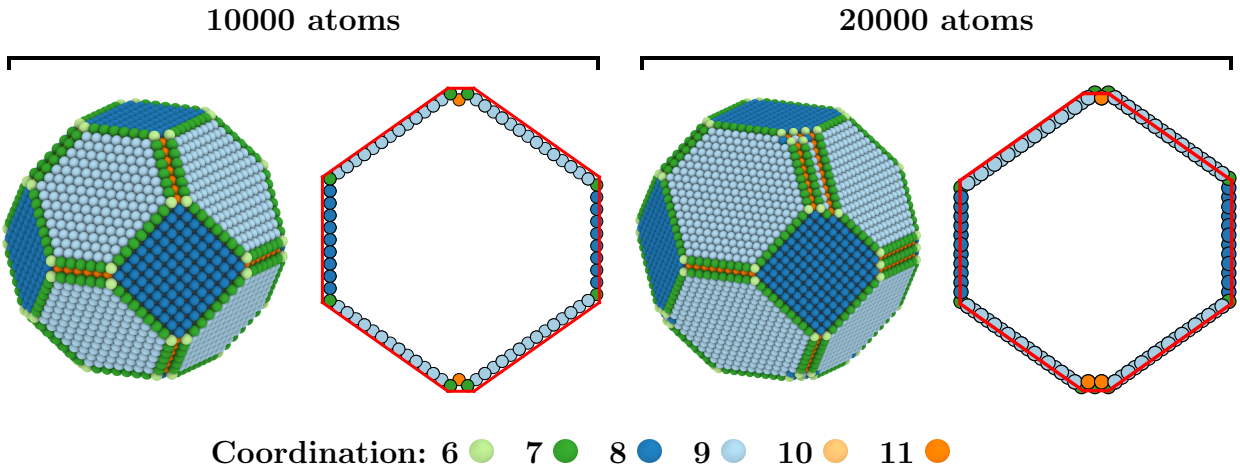


Figure 8: Low energy configurations using a linear approximation to the EMT potential for Ni. Here, $w_2 = 0.21w_1$. The optimization did not finish after 24 hours, but optimality gaps of 0.13% and 0.06% were established for the 10000 and 20000 atom systems respectively. The Wulff constructions have been superimposed on the surface atoms for a slice in the 110 direction, showing that the difference between the discrete and the continuum limit structures is small.

where $E(\sigma)$ is the energy of the system, $\mathbf{C}_n$ is the set of all n-body clusters, each containing cluster instances $\mathbf{c}_f$, $V_{n,f}$ is the effective cluster interaction (ECI) for the f^{th} n-body cluster instance, and σ_i is the pseudo-spin variable at each site i. There are many methods for choosing clusters, truncation of the expansion, and ECIs. Here we concern ourselves only with finding the ground state configuration of a chosen CE model.

A product of spin variables is a non-linear operation, which is forbidden in a MIP model. We handle this non-linearity by tranforming the spin variables $\sigma_i \in \{-1, 1\}$ to binary variables $x_i \in \{0, 1\}$ using the relation $\prod_{i\in\mathbf{C}} \sigma_i = \prod_{i\in\mathbf{C}} (2x_i - 1)$ which produces an equivalent Hamiltonian with different ECIs but only binary variables. A product of binary variables can be modelled using a generalization of the *logical AND* constraint used in Model 1 (equations (7)-(9)); for each cluster instance we introduce a binary variable $y_{\mathbf{C}}$ and two types of constraints: $y_{\mathbf{C}} \leq x_i \ \forall i \in \mathbf{C}$ and $y_{\mathbf{C}} \geq 1 - |\mathbf{C}| + \sum_{i\in\mathbf{C}} x_i$. Taken together, these constraints are equivalent to the relation $y_{\mathbf{C}} = \prod_{i\in\mathbf{C}} x_i$.

We have applied this MIP model to the three-body cluster expansion developed for 55-atom PdPt nanoparticles by Tan et al. [39] in order to study hydrogen adsorption energies. They have used ab-initio energy calculations to fit the ECIs, in conjunction with simulated annealing to find ground-state configurations. At every composition, the MIP model found the ground state in under 30 seconds on a laptop computer. Figure 9 shows the energy of formation of the true ground states and those found by simulated annealing. For four of the ten available nanoparticles, the configuration found by simulated annealing is not the true ground state, which highlights the difficulties faced

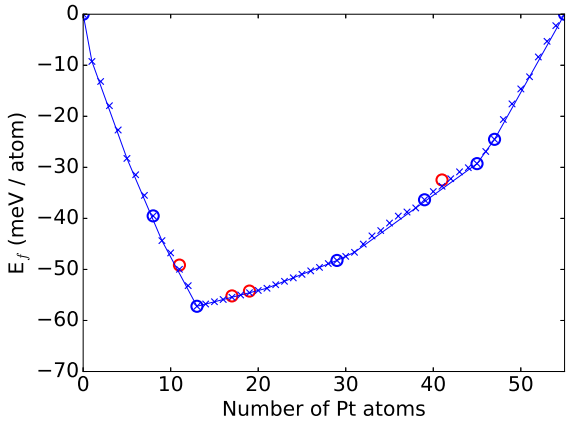


Figure 9: Energy of formation at every composition in a 55-atom PdPt nanoparticle in a cluster expansion model. The convex hull (blue line) shows the stable configurations. The ground state configurations (blue crosses) are found using a MIP model. Of ten configurations found using simulated annealing [39], six are the correct ground states (blue circles) and four are incorrect (red circles).

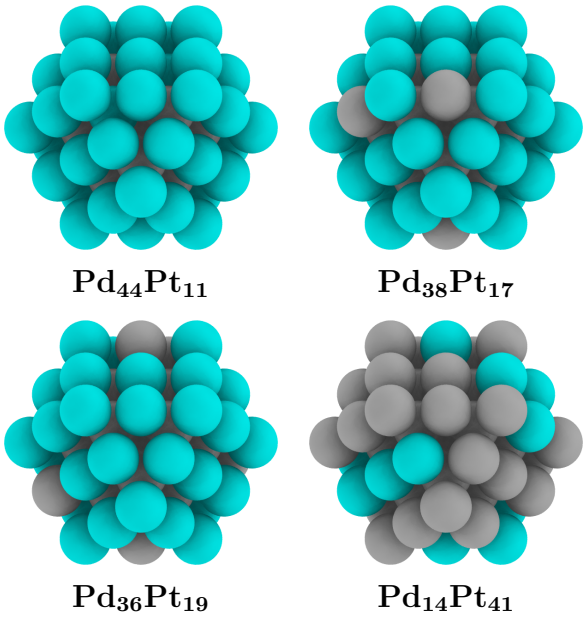


Figure 10: Correct ground state configurations of the four nanoparticles for which simulated annealing [39] did not find the lowest energy configuration.

when using stochastic methods, even at small system sizes.

5. Increasing the Potential Complexity

Whilst CE models are well studied and widely used, many authors nonetheless prefer to develop original energetic expressions, often in order to model complex behaviour which would otherwise require a very large CE basis. Recently, Kozlov et al. [45] developed a novel topological potential for use in determining chemical ordering in nanoparticles with ordered layers. The form of the potential is given by:

$$\begin{aligned} E_{\mathrm{TOP}} = E_0 + \epsilon^{\mathrm{A\text{-}B}}_{\mathrm{BOND}} N^{\mathrm{A\text{-}B}}_{\mathrm{BOND}} + \epsilon^{\mathrm{A}}_{\mathrm{CORNER}} N^{\mathrm{A}}_{\mathrm{CORNER}} + \epsilon^{\mathrm{A}}_{\mathrm{EDGE}} N^{\mathrm{A}}_{\mathrm{EDGE}} \\ + \epsilon^{\mathrm{A}}_{\mathrm{TERRACE}} N^{\mathrm{A}}_{\mathrm{TERRACE}} + \epsilon_{\mathrm{LAYER}} N_{\mathrm{LAYER}} \end{aligned} \tag{13}$$

Here, A and B denote the two elemental types in the nanoparticle. The potential is a function of the number of heteroatomic bonds $\left(N^{\mathrm{A\text{-}B}}_{\mathrm{BOND}}\right)$, the number of A atoms at corner, edge and terrace sites ($N^{\mathrm{A}}_{\mathrm{CORNER}}$, $N^{\mathrm{A}}_{\mathrm{EDGE}}$, $N^{\mathrm{A}}_{\mathrm{TERRACE}}$), and a layering term to account for tetragonal lattice distortion, given by $N_{\mathrm{LAYER}} = \sum_{\mathrm{k}} \left| n^{\mathrm{A}}_k - n^{\mathrm{B}}_k \right|$. The missing terms, such as $N^{\mathrm{B}}_{\mathrm{CORNER}}$, are uniquely determined by the terms included in the potential and are accounted for by the constant term E_0. The parameters for each term are fitted to ab initio energy calculations. Whilst the authors use stochastic search to find low energy configurations, this is an ideal application for a MIP model. Interestingly, if

the layer terms are excluded, the model is a simple two-body CE. As we showed in the previous section, these can be modelled using binary variables only. The layer terms necessitate the use of integer variables as well as more advanced MIP modelling concepts.

Model 2 shows the model we have developed for finding provably optimal ground states for the above potential. Each predetermined site with index i (14) has an associated binary variable x_i (17) which determines whether the atom at that site is an A-type or a B-type atom. The system has a fixed stoichiometry, which we enforce by fixing the number of B-type atoms (26). Each site has a cost associated with placing an A-type atom at that site (21), which either take a value equal to one of $\epsilon^{\mathrm{A}}_{\mathrm{CORNER}}$, $\epsilon^{\mathrm{A}}_{\mathrm{EDGE}}$, or $\epsilon^{\mathrm{A}}_{\mathrm{TERRACE}}$, or zero for an interior site. Bonds exist between every pair of sites. There are many ways of relating atom types to bond types. A simple way is:

$$x_i + x_j = y^{\mathrm{AB}}_{ij} + 2y^{\mathrm{BB}}_{ij} \quad \forall i,j \quad \text{and} \quad y^{\mathrm{AA}}_{ij} + y^{\mathrm{AB}}_{ij} + y^{\mathrm{BB}}_{ij} = 1 \quad \forall i,j \tag{34}$$

which captures all bonds types. Alternatively, since only AB bonds are required for this energetic model, logical XOR constraints [35] could be used instead. However, through experimentation we have found that the most efficient formulation is to use a combination of BB bonds and site multipliers. A BB bond variable is defined between each pair of sites (18) and is related to the site types by constraints (27)-(29). Each AB bond has an energy (22) which is equal to $\epsilon^{\text{A-B}}_{\mathrm{BOND}}$ for nearest-neighbour bonds and zero otherwise. The lower line of the objective function (25) combines the site types and the BB bond variables to capture the AB bond energy contribution.

The layering term is relatively complex; each layer has an index (15), a decision variable (19), and two absolute-value variables (20). The decision variable is used to permit only one of either Δ^{A}_k or Δ^{B}_k to take a nonzero value, using the indicator constraints (31)-(32). These are used in conjunction with equation (30), which is a standard formulation for calculating absolute values. The terms for site energies, bond energies and layers energies are collected in the objective function (25). Lastly, the performance of the model can be greatly improved by adding symmetry breaking constraints [46]. Since the structures we are optimizing have cubic symmetry (c.f. Figure 11), for every configuration there exists at least 47 equivalent solutions. Symmetry in MIP models drastically increases solution time, since it causes an explosion in the number of branches which must be explored. Our symmetry constraints impose an ordering on the site variables. Each symmetry constraint with index g (16), consists of two sets of site variables (24), and an ordering relation (33). We define three symmetry constraints, each of which splits the nanoparticle into two sets based on which side of the origin they lie, for each of the x, y and z axes.

We have used the parameters provided by Kozlov et al. to optimize the chemical ordering of nanoparticles with four different elemental compositions, at a range of sizes. The resulting ground-state configurations for the largest nanoparticles are shown in Figure 11. The running times are shown in Figure 12. The layering terms in the PdZn systems require extra variables and constraints, and cause the model to solve more slowly. The PdCu nanoparticles highlight the variable performance of solving a MIP

Indices:	$i, j \in \mathbf{I}$		Site index	(14)
	$k \in \mathbf{K}$		Layer index	(15)
	$g \in \mathbf{G}$		Symmetry constraint index	(16)
Variables:	$x_i \in \{0, 1\}$	$\forall i$	Element type (A/B) at site i	(17)
	$y_{ij}^{\mathrm{BB}} \in \{0, 1\}$	$\forall i, j$	BB bonds inactive/active	(18)
	$w_k \in \{0, 1\}$	$\forall k$	Decision variable for each layer	(19)
	$\Delta_k^{\mathrm{A}}, \Delta_k^{\mathrm{B}} \in \mathbb{N}^0$	$\forall k$	Layer absolute-value variables	(20)
Parameters:	$E_i \in \mathbb{R}$	$\forall i$	Cost of A-type atom at site i	(21)
	$E_{ij}^{\mathrm{AB}} \in \mathbb{R}$	$\forall i, j$	Site i-j bond energy	(22)
	$\mathbf{L}_k$	$\forall k$	Indices of sites in each layer	(23)
	$\mathbf{U}_g, \mathbf{V}_g$	$\forall g$	Symmetry constraint sites	(24)
Minimize:	$\sum_i E_i (1 - x_i) + \sum_k E_k \left(\Delta_k^{\mathrm{A}} + \Delta_k^{\mathrm{B}}\right)$		Total energy of system	(25)
	$+ \sum_{ij} E_{ij}^{\mathrm{AB}} (x_i + x_j) - 2 \sum_{ij} E_{ij}^{\mathrm{AB}} y_{ij}^{\mathrm{BB}}$			
Subject to:	$\sum_i x_i = N_{\mathrm{B}}$		Fixed number of B-type atoms	(26)
	$y_{ij} \leq x_i$	$\forall i, j$	BB bond constraint	(27)
	$y_{ij} \leq x_j$	$\forall i, j$	BB bond constraint	(28)
	$y_{ij} \geq x_i + x_j - 1$	$\forall i, j$	BB bond constraint	(29)
	$\Delta_k^{\mathrm{A}} - \Delta_k^{\mathrm{B}} = 2 \sum_{i \in \mathbf{L}_k} x_i - \lvert\mathbf{L}_k\rvert$	$\forall k$	Layer term absolute value	(30)
	$\Delta_k^{\mathrm{A}} \leq 2 \lvert\mathbf{L}_k\rvert w_k$	$\forall k$	Layer indicator constraint	(31)
	$\Delta_k^{\mathrm{B}} \leq 2 \lvert\mathbf{L}_k\rvert (1 - w_k)$	$\forall k$	Layer indicator constraint	(32)
	$\sum_{i \in \mathbf{U}_g} x_i \geq \sum_{i \in \mathbf{V}_g} x_i$	$\forall g$	Symmetry breaking constraint	(33)

Model 2: A model for minimizing the energy in a bimetallic nanoparticle, using a topological potential. Here, the lattice sites are fixed, and the model finds the optimal placement of A-type and B-type atoms. This is a more general model than needed for optimizing energy given by equation 13.

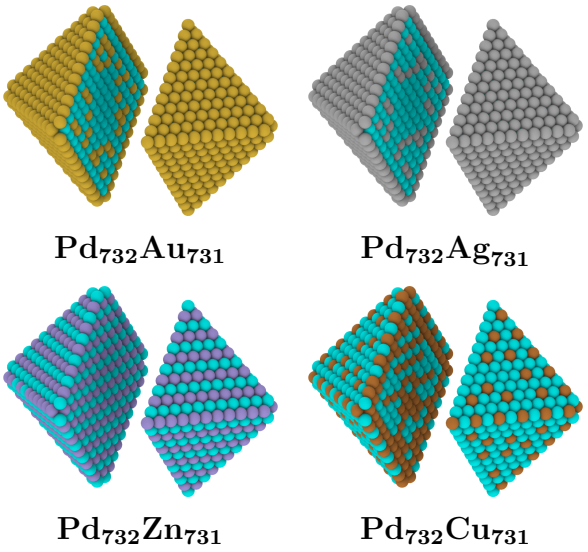


Figure 11: Ground state configurations for bimetallic nanoparticles using the potential defined by Kozlov et al. [45] (c.f. equation 13)

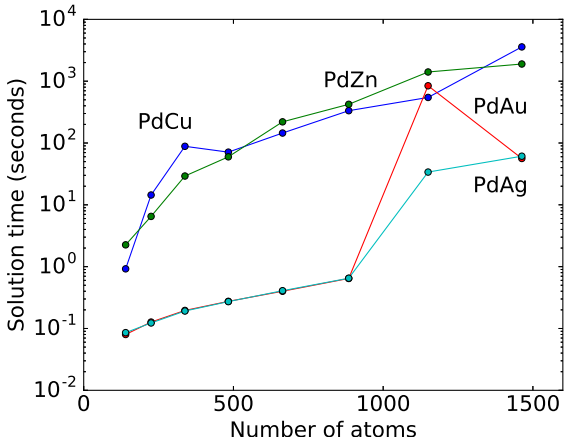


Figure 12: Solution times for bimetallic nanoparticles using a topological potential, for different nanoparticle sizes. The times were measured using 8 threads on a 2.80GHz Intel Xeon E5-2680 CPU with 128GB RAM using the Gurobi 6.5 solver [47].

model. The branch and bound algorithm is most effective when different solutions have a large variability in cost; this allows aggressive pruning of the search tree. The PdCu system is different from the others in that it exhibits a strong preference for mixing, rather than segregation. Consequently, there are many solutions with similar costs. Since every structure was solved in under an hour, we argue that for systems of moderate size the approach is superior to stochastic search methods. We note that the obtained exact configurations (c.f. Figure 11) are qualitatively similar to the configurations found by Kozlov et al. using Monte Carlo methods. However, the configurations differ in the details. For example, the exact configurations exhibit more symmetry than those obtained with MC.

6. Conclusions

XXXX

Acknowledgements

P.M. Larsen thanks J. Larsen advice on selecting solver settings. This work was supported by a research grant (9455) from VILLUM FONDEN.

References

[1] Smith B E 2002 *Science* **297** 1654–1655

[2] Dahl S, Logadottir A, Egeberg R, Larsen J, Chorkendorff I, Tornqvist E and Norskov J 1999 *Phys. Rev. Lett.* **83** 1814–1817

[3] Honkala K, Hellman A, Remediakis I, Logadottir A, Carlsson A, Dahl S, Christensen C and Norskov J 2005 *Science* **307** 555–558

[4] Lopez N, Janssens T, Clausen B S, Xu Y, Mavrikakis M, Bligaard T and Nørskov J 2004 *J Catal* **223** 232–235

[5] Hvolbæk B, Janssens T, Clausen B S, Falsig H, Christensen C and Nørskov J 2007 *Nano Today* **2** 14–18

[6] Gavnholt J and Schiøtz J 2008 *Phys. Rev. B* **77** 35404

[7] Brodersen S H, Grønbjerg U, Hvolbæk B and Schiøtz J 2011 *J Catal* **284** 34–41

[8] Lysgaard S, rdal J x n S G M x, Hansen H A and Vegge T 2015 *Phys Chem Chem Phys* **17** 28270–28276

[9] Hartke B 1993 *The Journal of Physical Chemistry* **97** 9973–9976

[10] Lee J, Lee I H and Lee J 2003 *Phys. Rev. Lett.* **91** 080201–4

[11] Wales D J and Scheraga H A 1999 *Science* **285** 1368–1372

[12] Goedecker S 2004 *J. Chem. Phys.* **120** 9911–9917

[13] Rossi G and Ferrando R 2009 *Journal of Physics: Condensed Matter* **21** 084208

[14] Burke E K, De Causmaecker P, Berghe G V and Van Landeghem H 2004 *Journal of scheduling* **7** 441–499

[15] Santos H G, Toffolo T A, Gomes R A and Ribas S 2016 *Annals of Operations Research* **239** 225–251

[16] Delgado A, Jensen R M, Janstrup K, Rose T H and Andersen K H 2012 *European Journal of Operational Research* **220** 251–261

[17] Ehrgott M and Ryan D M 2002 *Journal of Multicriteria Decision Analysis* **11** 139

[18] Butchers E R, Day P R, Goldie A P, Miller S, Meyer J A, Ryan D M, Scott A C and Wallace C A 2001 *Interfaces* **31** 30–56

[19] Lysgaard J, Letchford A N and Eglese R W 2004 *Mathematical Programming* **100** 423–445

[20] Toth P and Vigo D 2014 *Vehicle routing: problems, methods, and applications* (SIAM)

[21] Cheng C, Feiring B and Cheng T 1994 *International Journal of Production Economics* **36** 291–305

[22] Vance P H 1998 *Computational optimization and applications* **9** 211–228

[23] Padberg M and Rinaldi G 1991 *SIAM review* **33** 60–100

[24] De Simone C, Diehl M, Jünger M, Mutzel P, Reinelt G and Rinaldi G 1995 *Journal of Statistical Physics* **80** 487–496

[25] Hartmann A K and Rieger H 2006 *New optimization algorithms in physics* (John Wiley & Sons)

[26] Yang R, Dai J, Yang Y and Hu Y 2006 *Physics in Medicine and Biology* **51** 3653

[27] Collet G, Andonov R, Yanev N and Gibrat J F 2011 *Discrete Applied Mathematics* **159** 1707–1716

[28] Huang W, Kitchaev D A, Dacek S T, Rong Z, Urban A, Cao S, Luo C and Ceder G 2016 *Phys. Rev. B* **94** ISSN 2469-9950

[29] Nemhauser G L and Wolsey L A 1988 *Integer Programming and Combinatorial Optimization* (John Wiley & Sons)

[30] Chen D S, Batson R G and Dang Y 2010 *Applied integer programming: modeling and solution* (John Wiley & Sons)

[31] Baletto F, Ferrando R, Fortunelli A, Montalenti F and Mottet C 2002 *The Journal of chemical physics* **116** 3856–3863

[32] Daw M S and Baskes M I 1984 *Physical Review B* **29** 6443

[33] Jacobsen K, Stoltze P and Nørskov J 1996 *Surface science* **366** 394–402

[34] Rosato V, Guillope M and Legrand B 1989 *Philosophical Magazine A* **59** 321–336

[35] Williams H P 2013 *Model building in mathematical programming* (John Wiley & Sons)

[36] Larsen P M, Jacobsen K W and Schiøtz J 2017 Computational Materials Repository `http://cmr.fysik.dtu.dk/`

[37] Van Hardeveld R and Hartog F 1969 *Surface Science* **15** 189–230

[38] Sanchez J, Ducastelle F and Gratias D 1984 *Physica A* **128** 334–350 ISSN 0378-4371
[39] Tan T L, Wang L L, Johnson D D and Bai K 2012 *Nano Letters* **12** 4875–4880 ISSN 1530-6984
[40] Wang L L, Tan T L and Johnson D D 2014 *Nano Letters* **14** 7077–7084 ISSN 1530-6984
[41] Chepulskii R V, Butler W H, van de Walle A and Curtarolo S 2010 *Scr. Mater.* **62** 179–182 ISSN 1359-6462
[42] Teeriniemi J, Melander M, Lipasti S, Hatz R and Laasonen K 2017 *J. Phys. Chem. C* **121** 1667–1674 ISSN 1932-7447
[43] Mueller T 2012 *Phys. Rev. B* **86** ISSN 1098-0121
[44] Mueller T and Ceder G 2010 *ACS Nano* **4** 5647–5656 ISSN 1936-0851
[45] Kozlov S M, Kovács G, Ferrando R and Neyman K M 2015 *Chemical Science* **6** 3868–3880
[46] Fahle T, Schamberger S and Sellmann M 2001 Symmetry breaking *International Conference on Principles and Practice of Constraint Programming* (Springer) pp 93–107
[47] Gurobi Optimization I 2016 Gurobi optimizer reference manual URL `http://www.gurobi.com`

Paper 5

Alloy Design as an Inverse Problem of Cluster Expansion Models
Larsen P M, Kalidindi A R, Schmidt S and Schuh C A
Acta Materialia
Volume 139

Acta Materialia xxx (2017) 1–7

Contents lists available at ScienceDirect

Acta Materialia

journal homepage: www.elsevier.com/locate/actamat

Full length article

Alloy design as an inverse problem of cluster expansion models

Peter Mahler Larsen [a, b], Arvind R. Kalidindi [a], Søren Schmidt [b], Christopher A. Schuh [a, *, 1]

[a] Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[b] Department of Physics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Article history:
Received 20 June 2017
Received in revised form
3 August 2017
Accepted 4 August 2017
Available online xxx

Keywords:
Cluster expansion
Computational thermodynamics
Alloys
Microstructure design
Phase diagrams

ABSTRACT

Central to a lattice model of an alloy system is the description of the energy of a given atomic configuration, which can be conveniently developed through a cluster expansion. Given a specific cluster expansion, the ground state of the lattice model at 0 K can be solved by finding the configuration of solutes that minimizes the energy of the system. In this paper, we develop a method for solving the inverse lattice problem, where, given a broad class of potential, we find the ground states for *all possible* values of the effective cluster interaction energies. To do so, we formulate the inverse problem in terms of energetically distinct configurations, using a constraint satisfaction model to identify constructible configurations, and show that a convex hull can be used to identify ground states. To demonstrate the approach, we solve for all ground states for a binary alloy in a 2D hexagonal lattice both with and without an interface, based on pairwise interactions.

1. Introduction

Statistical mechanics models of alloys assign probabilities to the possible configurations of alloying elements and, based on a Maxwell-Boltzmann distribution, determine the equilibrium state of the alloy. Describing the configuration space of the alloy inevitably requires some approximation. The simplest models rely on assumptions of random distributions of each alloying element, either in the entire system, as in an ideal solution model, or within any phase as in a regular solution model. A major improvement to capturing configurational degrees of freedom is a lattice model, also known as the generalized Ising model [1] or Cluster Expansion [2] (CE) model, where each lattice site represents a single atom of a particular element with a pseudo-spin occupancy variable σ_i that, for the binary case, is -1 or $+1$ for a solute or solvent atom, respectively. The energy of an alloy configuration, $\boldsymbol{\sigma}$, in the lattice model with a cluster expansion interatomic potential$\mathbf{V}$, can generally be defined by the Hamiltonian [3]:

$$H(\boldsymbol{\sigma}, \mathbf{V}) = V_0 + \sum_{c \in \mathbf{C}} V_c \boldsymbol{\sigma}_c \tag{1}$$

where $\mathbf{C}$ is the set of clusters for which effective cluster interactions (ECIs), V_c, are prescribed. $\boldsymbol{\sigma}_c$ is a generic cluster function defined as the product of σ_i over all sites in a cluster and captures the solute configuration of the cluster. While a fixed site geometry still constrains the configuration space considered, lattice models are a remarkably accurate tool for developing phase diagrams of alloys, where a cluster expansion can be performed from ab-initio calculations [2–13].

The 0 K phase diagram is determined by the alloy configurations that minimize Equation (1) at different solute concentrations. The problem of merely solving for the minimum energy configurations we term the 'forward problem'. Conversely, the much more complex 'inverse problem' is the one underlying alloy design, where, given a desired alloy configuration the objective is to determine which alloy systems, if any, possess this configuration as an equilibrium state. The possible ground states of the alloy model are those for which there exists a parameter set $\mathbf{V}$ that minimizes energy according to Equation (1). While the relationship between energy and the cluster functions is linear, finding all possible ground states is difficult due to the constraints imposed on the cluster functions by the lattice, namely the integrality constraint of equalling -1 or 1.

Allen and Cahn were able to solve this inverse problem for body-centered cubic and face-centered cubic lattices where the interatomic potential is constrained to nearest-neighbor and next-nearest-neighbor pairwise bonds [14]. Their approach and the more general polytope method [15–18] circumvent the integrality

* Corresponding author.
E-mail address: schuh@mit.edu (C.A. Schuh).
[1] Prof. Christopher A. Schuh was an editor of the journal during the review period of the article. To avoid a conflict of interest, Prof. J. Rickman acted as editor for this manuscript.

http://dx.doi.org/10.1016/j.actamat.2017.08.008
1359-6454/

constraint by first selecting one base cluster (e.g. an octahedron for the body-centered cubic lattice), and then determining all possible ways in which the alloying elements can occupy sites in this cluster, assigning each configuration on the cluster a probability weight from 0 to 1. Since probability is a continuous function, this problem can be solved by standard linear programming techniques to determine all possible equilibrium states.

However, not all combinations of probabilities for each cluster are constructible in a larger lattice, and thus additional rules must be defined to ensure that a ground state determined in this way represents a physical system. In a crystalline phase with short range interactions, these rules can be reasonably simple, but in more complex systems constructibility can be a cumbersome problem [18]. Inverse problems in certain disordered systems, such as hyperuniform glasses, have been studied to determine the interatomic potentials that lead to disordered ground states [19,20], but are beyond the scope of a standard cluster expansion representation. There has been growing interest in studying chemical ordering at interfaces, such as grain boundaries, and constructing phase diagrams (also known as complexion diagrams) of the segregation and ordering at these interfaces [21–30]. Such problems have an inhomogeneous cluster expansion, since clusters at an interface have different energies than ones in the crystal, which makes use of the polytope method more challenging. Even in crystalline phases, considering clusters with longer range interactions can be challenging within the polytope method as a larger base cluster must be chosen leading to a larger number of potential configurations and more complex constructibility considerations.

The forward problem of finding ground states in complex systems has recently been formulated by Huang et al. [31] as a class of problem known as pseudo-Boolean optimization [32] (PBO). PBO models, which consist of an objective function to be optimized and a set of constraints to be satisfied, greatly improve the generality and speed of solving lattice models given a cluster expansion. In this paper, we present a framework for solving the inverse problem of the lattice model directly in the configuration space of the lattice using a constraint satisfaction model. We then investigate the ground state ordered states in a 2D hexagonal lattice both with and without an interface to demonstrate how all possible configurations with a given potential form can be calculated.

2. The inverse problem in lattice models

The forward problem asks: for a specified $\mathbf{V}$, what is the ground state configuration? A configuration $\boldsymbol{\sigma}$ is a ground state if it has a lower energy than all other configurations:

$$H(\boldsymbol{\sigma},\mathbf{V}) < H\left(\boldsymbol{\sigma}',\mathbf{V}\right) \quad \forall \boldsymbol{\sigma}' \neq \boldsymbol{\sigma} \tag{2}$$

Then, the inverse problem asks: for a specified configuration $\boldsymbol{\sigma}$, does there exist a set $\mathbf{V}$ for which $\boldsymbol{\sigma}$ is the ground state? Here, the space of interactions being considered (e.g. nearest neighbor, next-nearest neighbor, etc.) is constant. We denote such a configuration as *minimizing*. It must satisfy the condition:

$$\exists \mathbf{V} \quad \text{s.t.} \quad H(\boldsymbol{\sigma},\mathbf{V}) < H\left(\boldsymbol{\sigma}',\mathbf{V}\right) \quad \forall \boldsymbol{\sigma}' \neq \boldsymbol{\sigma} \tag{3}$$

Thus, to solve the inverse problem, all possible ECIs as well as all possible configurations must be considered in order to find the ground states. In order to make this tractable, we will first reduce the configuration space to a smaller, abstract space, and then use arguments based on principles of convexity to identify minimizing configurations.

In order to reduce the configuration space, we note that the Hamiltonian in Equation (1) can be expanded to lattice-gas [33] form:

$$H(\mathbf{x},\mathbf{E}) = E_0 + \sum_j \sum_{\mathbf{c}\in\mathbf{C}_j} \prod_{i\in\mathbf{c}} E_j x_i \tag{4}$$

where spin variables $\sigma_i \in \{-1,1\}$ of the cluster function have been replaced by a binary variable $x_i \in \{0,1\}$ using the relation $\sigma_i = 2x_i - 1$, and $\mathbf{E}$ is the equivalent set of ECIs. For each cluster instance we can then define a binary variable $y_{\mathbf{c}} = \prod_{i\in\mathbf{c}} x_i$ which denotes *cluster instance activity*, where 0 and 1 denote an inactive and active cluster instance respectively. Using Equation (4), the Hamiltonian can then be rewritten in the number of active cluster instance (ACI) counts:

$$H(\mathbf{x},\mathbf{E}) = \langle \mathbf{E},\mathbf{N}\rangle \tag{5}$$

where $\mathbf{N} = \{\sum_{\mathbf{c}\in\mathbf{C}_1} y_{\mathbf{c}}, \sum_{\mathbf{c}\in\mathbf{C}_2} y_{\mathbf{c}}, \sum_{\mathbf{c}\in\mathbf{C}_3} y_{\mathbf{c}}, \ldots\}$ is the vector of ACI counts (ACI vector). Representing a configuration as an ACI vector is a much more compact description and permits a simpler expression of the Hamiltonian. Using this parametrization, we define the 'energy space' $V(\mathbf{N})$ of a configuration as the region of ECI space for which it is a ground state:

$$V(\mathbf{N}) = \{\mathbf{E} |\ \langle \mathbf{E},\mathbf{N}\rangle < \langle \mathbf{E},\mathbf{N}'\rangle \ \forall\ \mathbf{N}' \neq \mathbf{N}\} \tag{6}$$

where $\mathbf{N}$ and $\mathbf{N}'$ denote constructible bond count vectors with the same number of solute atoms.

Determination of the minimizing configurations proceeds by identifying the convex hull [34] of configurations. The *maximum principle* [35] states that the maximum (or equally, minimum) of any convex function on a compact convex set is attained at the boundary of the set. Thus, in order to exploit the maximum principle, we require a convex function of a convex compact set. If we relax the implicit integrality constraints on $\mathbf{N}$, the inner product in Equation (5) is a sum of linear functions, which is a convex function. Next, let $\mathbf{S} = \{\mathbf{N}_1,\mathbf{N}_2,\mathbf{N}_3,\ldots\}$ be the set of all constructible ACI vectors for a given set of clusters. Then, the convex hull of S is by definition a compact convex set. By restricting the domain ($\mathbf{N}$) of the Hamiltonian in Equation (5) to the convex hull of $\mathbf{S}$, we satisfy the necessary conditions of the maximum principle. As such, we can state that all possible ground states for a given $\mathbf{E}$ must lie on the boundary of the convex hull of $\mathbf{S}$. For practical reasons we can tighten the definition: any minimizing configuration must lie on a vertex of the convex hull of $\mathbf{S}$. We can do this since any configuration which lies on a plane of the convex hull has a zero energy space according to Equation (6).

What the above means is that solving the inverse problem amounts to finding all states that lie on the vertices of the convex hull of cluster space, which is a space where each axis is the number of counts of a given cluster. We illustrate this concept with a 1D example. Fig. 1 shows a periodic lattice in $\mathbb{R}^1$, with bonds between adjacent sites only, and whose sites are occupied by either A-type or B-type atoms. Given this energetic model, there are two types of clusters: 1-body and nearest-neighbor 2-body clusters. For a fixed concentration of B-type atoms, though, the contribution of 1-body clusters is the same in each case, meaning there is only one linearly independent cluster. The ACI vector of any configuration in this model can therefore be written as a 1D vector, $\mathbf{N} = \{k_1\}$ where k_1 is the number of nearest-neighbor B–B bonds. As such, the convex hull of all constructible ACI vectors consists of two points. These states, which constitute the minimizing configurations of the lattice, correspond to preferences for heteroatomic bonds (Fig. 1a) and

Please cite this article in press as: P.M. Larsen, et al., Alloy design as an inverse problem of cluster expansion models, Acta Materialia (2017), http://dx.doi.org/10.1016/j.actamat.2017.08.008

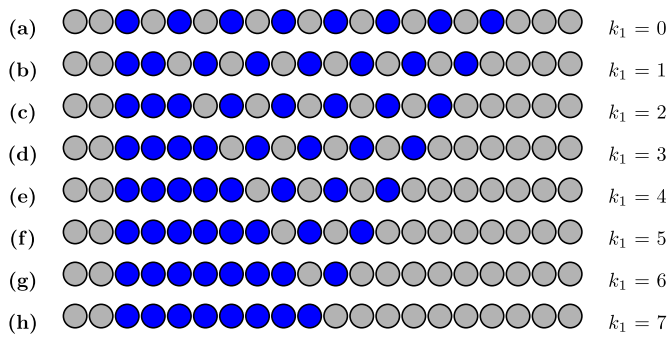


Fig. 1. Different configurations with the same stoichiometry in a periodic 1D lattice. Bonds exist between adjacent sites, which are occupied by either A-type (gray) or B-type (blue) atoms. The configurations shown here are energetically unique configurations for 12 A-type and 8 B-type atoms. Of the eight configurations, only two, **(a)** and **(h)**, are possible ground state configurations, as they have extreme values of k_1 (the number of nearest-neighbor B-B bonds). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

for homoatomic bonds (Fig. 1h).

3. Enumerating constructible configurations

In order to determine the convex hull of cluster space and find the set of minimizing configurations, we need to consider all possible ACI vectors. A naive method is to simply construct all possible configurations at a specified number of solute atoms, from which the set of ACI vectors can be obtained. This method, however, leads to many redundant calculations. Consider the 1D example in Section 2; with n_s lattice sites and n_b B-type atoms there are $\binom{n_s}{n_b}$ unique configurations, yet only n_b unique ACI vectors. For the illustrated example with $n_s = 20$ and $n_b = 8$ there are 125970 configurations of which only 8 have unique ACI vectors.

Instead, the ACI vectors can be enumerated directly, which amounts to considering only energetically distinct configurations. This approach poses a new challenge: not all ACI vectors are constructible. For example, for $n_b < n_s$ all ACIs with $k_1 \geq n_b$ are inconstructible. In the 1D example this constraint is obvious, but in higher dimensional lattices or lattices with different site types the conditions for constructibility are non-trivial.

Model 1
A constraint satisfaction model for determining constructibility of an ACI vector. The model determines whether site occupancies can be selected such that the constraints are not violated.

Variables:				
	$x_i \in \{0,1\}$	$\forall i$	Type of atom site i	(7)
	$y_{\mathbf{c}} \in \{0,1\}$	$\forall \mathbf{c}$	Cluster instance variable	(8)
Parameters:				
	$\mathbf{N} \in \mathbb{N}_0^{\lvert\mathbf{J}\rvert}$		ACI vector	(9)
	$n_b \in \mathbb{N}_0$		Number of B − type atoms	(10)
Constraints:				
	$\sum_i x_i = n_b$		Fixed stoichiometry	(11)
	$y_{\mathbf{c}} = \text{AND}(\{x_i \mid i \in \mathbf{c}\})$	$\forall \mathbf{c}$	Cluster instance activity	(12)
	$\sum_{\mathbf{c} \in \mathbf{C}} y_{\mathbf{c}} = \mathbf{N}_j$	$\forall \mathbf{C}$	ACI constraint	(13)

In a more complex CE model we can test the constructibility of an ACI vector with a constraint satisfaction model, as shown in Model 1. The binary site variables (7) are constrained to the desired stoichiometry (10) by Equation (11). The cluster activity variables (8) are related to the site variables by a logical AND constraint (12), which is equivalent to the relation $y_{\mathbf{c}} = \prod_{i \in \mathbf{c}} x_i$. Lastly, the site activity constraints are related to the ACI vector (9) by Equation (12). The model is satisfiable if and only if the ACI is constructible. By using standard translations [36,37] for the logical AND constraint in Equation (13), the model can be formulated either as a pseudo-Boolean satisfiability problem [37] or a Mixed Integer Programming [38,39] (MIP) problem. For this application the methods are equivalent, and both can quickly determine the satisfiability of the model. In this work we have used the Gurobi 7.0 MIP solver [40].

Algorithm 1 For finding all minimizing configurations of a CE model.

Input:	Site geometry, clusters, desired stoichiometry.
1	Identify linearly independent set of clusters
2	Enumerate all ACI vectors
3	Determine constructibility of remaining ACI vectors using Model 1.
4	Calculate convex hull of constructible ACI vectors
Return:	Vertices of convex hull (minimizing configurations)

The method for determining all minimizing configurations is summarized in Algorithm 1. It is intended as a general recipe rather than a strict set of guidelines. Indeed, a variety of application specific optimizations can be applied at each step. For example, if an ACI can be decomposed into subsets, it may be advantageous to test constructibility (step 3) of the subsets separately. Standard convex hull algorithms may be used to maintain a boundary set of ACI vectors, which may permit entire regions of non-boundary ACI vectors to be discarded. Since these optimizations are inherently application specific, we do not discuss them in detail. Rather, we describe some chosen optimizations for the applications presented in the next sections.

This approach provably finds all possible ground states given a choice of interatomic potential class. In addition, the approach is flexible to consider complex interatomic potentials and can be implemented using commercial solvers for constraint satisfaction problems.

4. Application to a hexagonal lattice

First, we demonstrate the approach in a very simple model system which is still non-trivial: determining the minimizing configurations in a 2D hexagonal lattice model with nearest-neighbor and next-nearest-neighbor pairwise bonds. We fix the number of solute atoms to 24 on a 12 × 12 lattice. The ACI vector contains two linearly independent parameters, k_1 and k_2, which denote the number of active clusters (the number of B–B bonds) of the nearest-neighbor and next-nearest-neighbor types respectively. Rather than test all possible ACI vectors for constructibility, a MIP model is used to identify the maximum and minimum values of k_1 for every value of k_2. By doing so, we limit the number of constructibility tests to a small region of ACI space.

Fig. 2 shows the space of constructible ACI vectors for the chosen lattice model. The ECIs determine an optimal direction in ACI space. A configuration which is extreme in the optimal direction is the ground state for the chosen ECIs. Here, the convex hull of the constructible vectors clearly identifies six minimizing

Please cite this article in press as: P.M. Larsen, et al., Alloy design as an inverse problem of cluster expansion models, Acta Materialia (2017), http://dx.doi.org/10.1016/j.actamat.2017.08.008

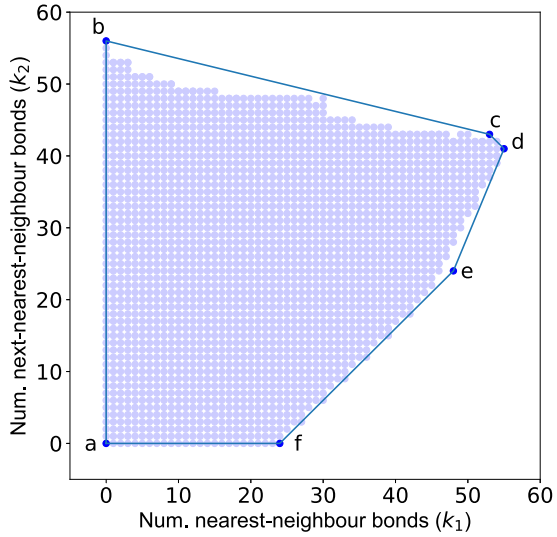


Fig. 2. All constructible ACI vectors (marked with a blue dot) in a 12×12 hexagonal lattice with nearest-neighbor and next-nearest-neighbor bonds and a fixed concentration of 24 solute atoms. The ACI vector $\mathbf{N} = \{k_1, k_2\}$ specifies the number of active clusters of each type. The convex hull is marked by a blue line; due to the *maximum principle*, each vertex of the convex hull represents a minimizing configuration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

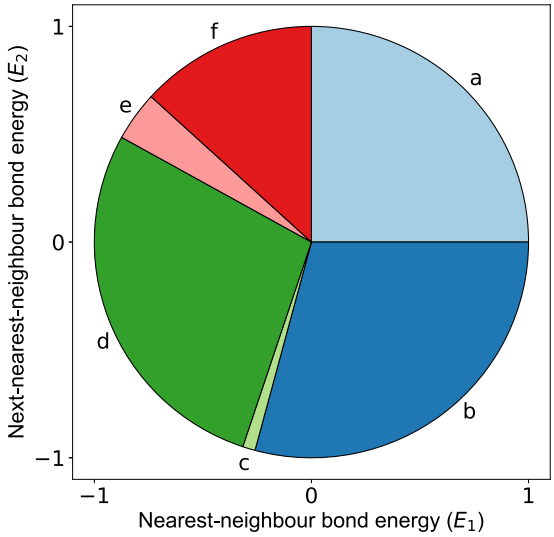


Fig. 3. Design map for the system. The sectors show the regions of ECI space for which each minimizing configuration is a ground state. All ECI vectors $\mathbf{E} = \{E_1, E_2\}$ are shown in the range $0 \leq |E| \leq 1$. Since the design map is a circular function, a single energetic parameter would suffice to describe any configuration.

configurations, which are the vertices of the convex hull. The minimizing configurations are also shown in energy space in Fig. 3. For a specified set of ECIs, the ground state configuration can be determined from this figure, which essentially constitutes a design map. In the general case, the boundary between two minimizing configurations, $\langle \mathbf{E}, \mathbf{N}_2 - \mathbf{N}_1 \rangle = 0$ (a hyperplane of equal energy), intersects the origin. Any design map is therefore a hyperspherical function, and all ECI vectors can be normalized without loss of generality. Thus, for a system with m linearly independent clusters, $m - 1$ energetic parameters are sufficient to describe any configuration.

Fig. 4 shows the solute configurations of the six minimizing configurations. The 'disordered mixing' configuration has a preference for heteroatomic bonding at both neighbor distances. A preference for heteroatomic nearest-neighbor bonds and homoatomic next-nearest-neighbor bonds produces the 'ordered mixing' configuration. The two bulk precipitation configurations exhibit preferences for homoatomic bonding at both neighbor distances, but in varying degrees; in the thermodynamic continuum limit, these configurations would also form a continuum of states. The 'lines' configuration requires a careful balance of bond preferences and, furthermore, exploits the periodic boundary conditions of the lattice to achieve an extreme ACI vector. Lastly, the 'triplets' configuration has a preference for homoatomic bonding at the nearest-neighbor distance only. These six states represent all possible ground state configurations in a nearest-neighbor and next-nearest neighbor pairwise bonding model on a 2D hexagonal lattice.

5. Application to a bicrystal in a hexagonal lattice

In addition to provably finding all minimizing configurations in a standard Ising model, this approach can be applied to more complex lattice problems, such as those containing interfaces where the cluster expansion is not consistent throughout the lattice. We consider the minimizing configurations at a grain boundary, a problem that we have studied previously using Monte Carlo simulations [41–44]. In this lattice model, a hexagonal lattice with a single grain boundary is considered, as shown in Fig. 5. The grain boundary in this 2D lattice is a line which bisects the lattice where any bonds between atoms on either side of this line are defined as grain boundary bonds (otherwise they are defined as intracrystalline). The model has a total of six different bond types; between each nearest-neighbor pairing of solvent and solute atoms (A and B) of either crystalline (c) or grain boundary (gb) type. At a fixed solute

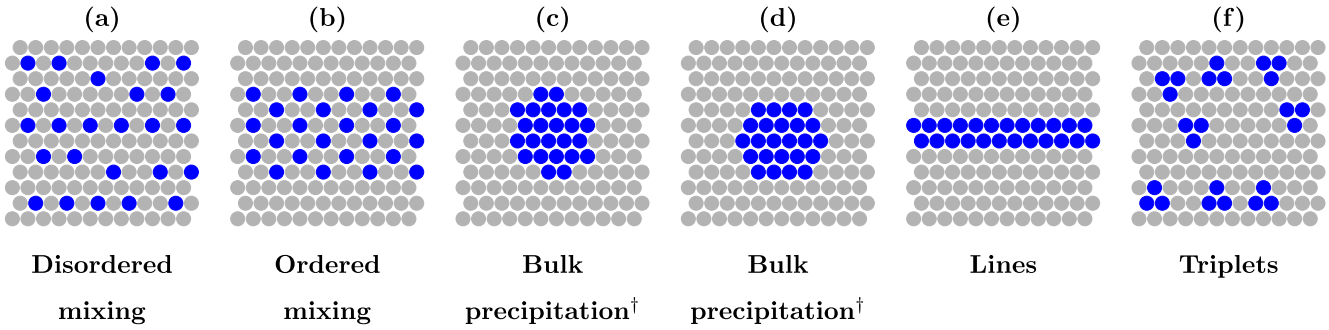


Fig. 4. All minimizing configurations for a 12×12 2D hexagonal lattice with 24 solute atoms. Each configuration lies at a vertex of the convex hull, shown in Fig. 2. †Bulk precipitation configurations form a continuum in the continuum limit; there are only two states here, due to the small number of solute atoms.

Please cite this article in press as: P.M. Larsen, et al., Alloy design as an inverse problem of cluster expansion models, Acta Materialia (2017), http://dx.doi.org/10.1016/j.actamat.2017.08.008

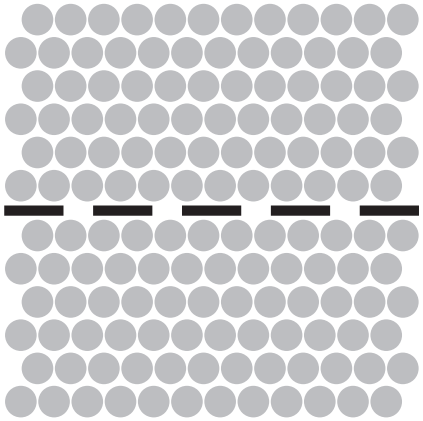


Fig. 5. Hexagonal bicrystal lattice studied for finding complexion types in 2D. The grain boundary, which lies along the [10] line, has been highlighted with a dashed line and by widening the gap between the lines of atoms. Periodic boundary conditions have been applied along the principal axes.

concentration, this bonding model has three linearly independent clusters; we can specify the ACI vector by $\mathbf{N} = \{n_{\mathrm{gb}}, k_{\mathrm{gb}}, k_{\mathrm{c}}\}$, where n_{gb} is the number of grain boundary sites occupied by solute atoms, and k_{gb} and k_{c} are the number of nearest-neighbor B–B bonds of the grain boundary and intracrystalline types respectively.

Our goal is to find all minimizing states of the 12×12 hexagonal lattice with 28 solute atoms (i.e. a concentration of ≈20% solute) under this nearest-neighbor bicrystal potential. This solute concentration is sufficiently large that the grain boundary can be entirely filled with solute atoms. At this concentration, the number of possible configurations is of the order 10^{30}. Thus, constructing each configuration directly is intractable, which motivates the use of the method described in the previous section. On the other hand, if we consider just the ACI vectors, the number of possible configurations is substantially reduced to 12337 possible ACI vectors for $n_{\mathrm{b}} = 28$, which is a small enough set to test for constructibility.

Using Model 1, we find that there are 7085 constructible ACI vectors. Note the very large increase in the complexity of the configuration space as compared with the interface-free hexagonal lattice; that system had only 2133 ACI vectors expressable in a 2D space, whereas the addition of the grain boundary requires approximately 3 times as many ACI vectors, which lie in a 3D space. The convex hull of these constructible ACI vectors contains exactly 12 vertices, as shown in Fig. 6. These are the minimizing configurations of the CE model, shown in Fig. 7. In a single-crystal model with nearest-neighbor pairwise bonds, only two states are formed: solute precipitation and solid solution. These states are also found here in the bicrystal model (bulk precipitation and bulk mixing); the addition of the grain boundary, even with only nearest neighbor interactions, introduces several new configurations that are ground states for some combination of energies for nearest neighbor bonds. Of these ground states, in particular several ordered states at the grain boundary, often called complexions [21], are found. The energies of the 6 bond types determine which ordered state forms, and in solving the inverse problem, a full mapping between the bond energy space and ground state configuration space is developed and can be read directly from the convex hull.

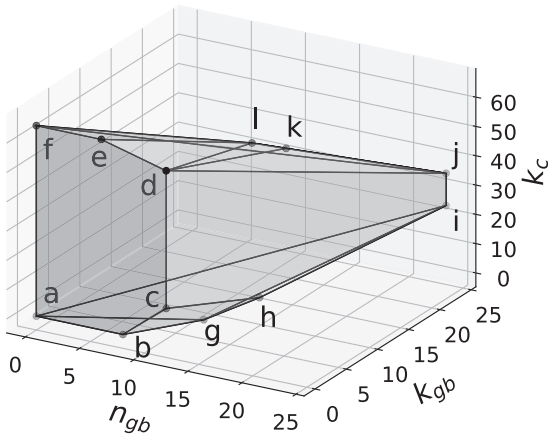


Fig. 6. All minimizing ACI vectors in a 12×12 hexagonal bicrystal lattice with nearest-neighbor bonds and a fixed concentration of 28 solute atoms. The ACI vector $\mathbf{N} = \{n_{\mathrm{gb}}, k_{\mathrm{gb}}, k_{\mathrm{c}}\}$ specifies the number of grain boundary sites occupied by solute atoms, and the number of B-B bonds of the grain boundary and intracrystalline types. Each vertex of the convex hull represents a minimizing configuration.

The ‘dots’ complexion type is characterized by a preference for A-B grain boundary bonds and a large penalty for B–B crystalline bonds, which leads to solute being separated along the grain boundary. An increase in the preference for A-B grain boundary bonds can bring more atoms to the grain boundary, producing a ‘repulsive line’ complexion. In this complexion, the preference for A-B grain boundary bonds is greater than the aversion to B–B crystalline bonds, meaning that B–B crystalline bonds will form, though only at the grain boundary. Changing the aversion to B–B crystalline bonds to a preference will produce an ‘attractive line’ complexion. Here, the preference for A-B grain boundary bonds is significantly stronger than the preference for B–B crystalline bonds. Reducing the preference for A-B grain boundary bonds results in a surface wetting complexion. In a 12×12 hexagonal lattice, only one surface wetting complexion exists, though in a larger lattice with more solute atoms the number of wetting complexions is larger, as the other two energetic parameters can change the shape of the precipitate. The surface wetting complexion shown here has the same number of B–B crystalline bonds as the bulk precipitation state, but with a weak preference for A-B grain boundary bonds that leads to its formation at the grain boundary.

The counterpart to ‘dots’ when B–B grain boundary bonds are preferred is the ‘dumbbells’ complexion, characterized by a preference for B–B grain boundary bonds and a large penalty for B–B crystalline bonds. Successive reductions in the penalty for B–B crystalline bonds again allow for greater grain boundary segregation, which results in the ‘armchairs’ and ‘repulsive lines’ complexion types respectively. Changing the penalty for B–B crystalline bonds to a preference produces an ‘attractive lines’ complexion type. Increasing the strength of this preference results in interior wetting. Trends in proclivities for particular types of bonds shown in Fig. 7 will be studied in more detail in future work.

Solving for ground states using the approach outlined in Algorithm 1 was well-suited to this problem. It is possible that we could have found all the minimizing configurations by sampling the bond energy space very finely, and finding the ground state configuration at each sample, though we would not have been able to guarantee that other ground states do not exist between samples. The approach presented here *provably* finds all minimizing configurations and the catalog in Fig. 7 is therefore complete. Alternatively, the polytope method could have been used to determine all possible ground states, but defining the compatibility between clusters at the grain boundary and clusters in the crystalline regions of the lattice would have been non-trivial. Model 1 allows for a non-homogeneous potential as in the case of the bicrystal lattice, and moreover by casting this problem in terms of constraint satisfaction, a MIP model could be used to find all the constructible states.

The application presented here is a small CE model with

Please cite this article in press as: P.M. Larsen, et al., Alloy design as an inverse problem of cluster expansion models, Acta Materialia (2017), http://dx.doi.org/10.1016/j.actamat.2017.08.008

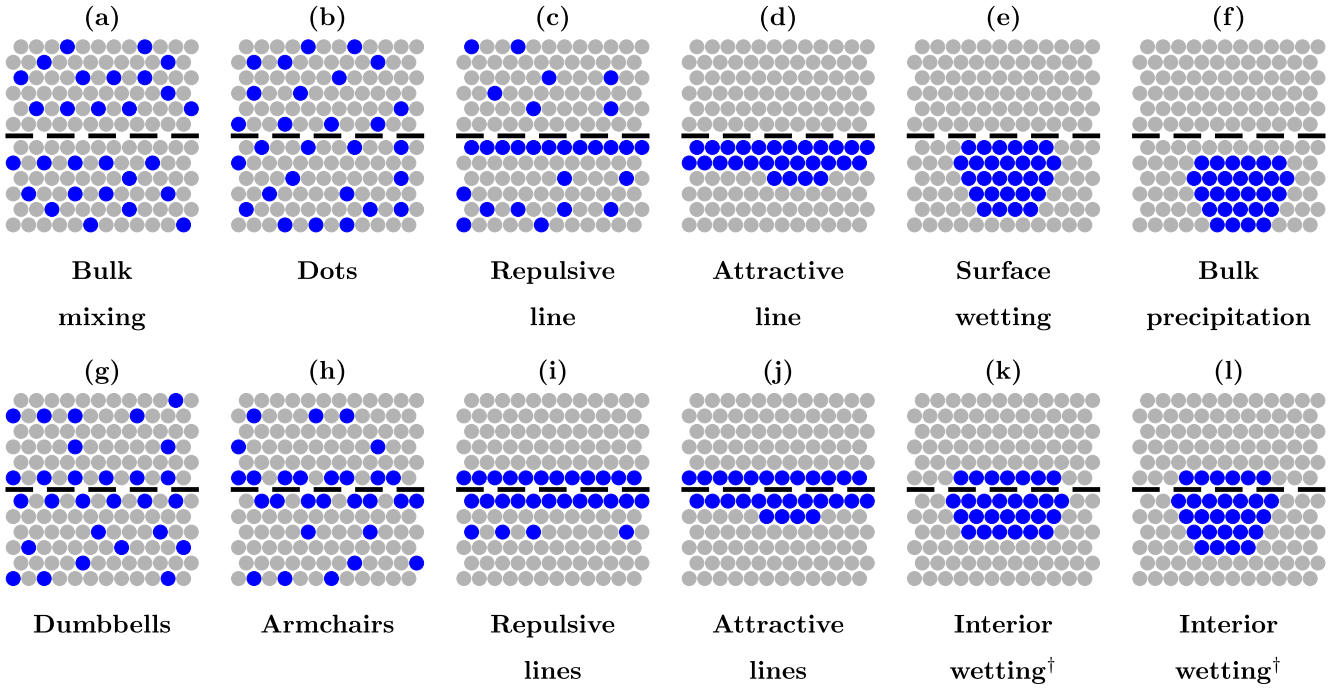


Fig. 7. All minimizing configurations for a 12×12 2D hexagonal lattice with 28 solute atoms, with complexions with a preference for A-B grain boundary bonds and B-B grain boundary bonds placed in separate rows. Solvent and solute atoms are colored gray and blue respectively. †Wetting complexions form a continuum as the precipitate can intersect the grain boundary in several ways; as such, we have not given them different names. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

effectively 2 clusters. However, enumeration of minimizing configurations is only of interest if the number of clusters is small, since this leads to a small number of minimizing configurations which can provide physical insight into the system being modelled. If the CE model contains many clusters, the number of minimizing configurations is large and most of them are not physically realizable and do not warrant consideration. Potential applications of this model include solving for all possible ground states in multicomponent alloys, compounds, and nanoparticles.

6. Conclusions

We have introduced a method for rigorously solving the inverse lattice problem to find all possible ground states given a specific class of cluster expansion. First we formalized the inverse problem for the cluster expansion model and showed that the inverse problem can be solved by determining all possible energetically distinct configurations, where all possible ground states lie on the convex hull in configuration space. To determine all distinct configurations, we considered the problem in cluster space, where counts of each type of cluster (termed an active cluster instance, ACI) are used to consider only configurations of different energy. The lattice model problem is cast into a constraint satisfaction problem to determine if it is possible to construct a configuration with a given set of ACI counts. Once configuration space is reduced to only constructible, energetically distinct configurations, the convex hull can be constructed and all possible ground states can be determined. To demonstrate this method, we determined all possible ground states in a bicrystal lattice model with nearest-neighbor interactions. In doing so, we showed how this method provably determines all possible ground states even in a non-homogeneous cluster expansion lattice problem. Software demonstrating the use of the method is available online [45].

Acknowledgements

This work was supported by the U.S. Army Research Office under grant W911NF-14-1-0539. P.M.L. received funding for an international research stay from the Otto Mønsteds Fond foundation (Grant number: 16-70-0049). A.R.K received funding from a National Defense Science and Engineering Fellowship.

References

[1] E. Ising, Beitrag zur Theorie des Ferromagnetismus, Z. Phys. A 31 (1) (1925) 253–258, http://dx.doi.org/10.1007/BF02980577.
[2] J. Sanchez, F. Ducastelle, D. Gratias, Generalized cluster description of multicomponent systems, Physica A 128 (1) (1984) 334–350, http://dx.doi.org/10.1016/0378-4371(84)90096-7.
[3] G. Ceder, A derivation of the Ising model for the computation of phase diagrams, Comput. Mater. Sci. 1 (2) (1993) 144–150, http://dx.doi.org/10.1016/0927-0256(93)90005-8.
[4] J.W.D. Connolly, A.R. Williams, Density-functional theory applied to phase transformations in transition-metal alloys, Phys. Rev. B 27 (8) (1983) 5169–5172, http://dx.doi.org/10.1103/PhysRevB.27.5169.
[5] K. Terakura, T. Oguchi, T. Mohri, K. Watanabe, Electronic theory of the alloy phase stability of Cu-Ag, Cu-Au, and Ag-Au systems, Phys. Rev. B 35 (5) (1987) 2169–2173, http://dx.doi.org/10.1103/PhysRevB.35.2169.
[6] S.-H. Wei, L.G. Ferreira, A. Zunger, First-principles calculation of temperature-composition phase diagrams of semiconductor alloys, Phys. Rev. B 41 (12) (1990) 8240–8269, http://dx.doi.org/10.1103/PhysRevB.41.8240.
[7] M. Asta, C. Wolverton, D. de Fontaine, H. Dreyssé, Effective cluster interactions from cluster-variation formalism. I, Phys. Rev. B 44 (10) (1991) 4907–4913, http://dx.doi.org/10.1103/PhysRevB.44.4907.
[8] M. Asta, R. McCormack, D. de Fontaine, Theoretical study of alloy phase stability in the Cd-Mg system, Phys. Rev. B 48 (2) (1993) 748–766, http://dx.doi.org/10.1103/PhysRevB.48.748.
[9] V. Ozoliņš, C. Wolverton, A. Zunger, Cu-Au, Ag-Au, Cu-Ag, and Ni-Au intermetallics: first-principles study of temperature-composition phase diagrams and structures, Phys. Rev. B 57 (11) (1998) 6427–6443, http://dx.doi.org/10.1103/PhysRevB.57.6427.
[10] M. Asta, J. Hoyt, Thermodynamic properties of coherent interfaces in f.c.c.-based AgAl alloys: a first-principles study, Acta Mater. 48 (5) (2000) 1089–1096, http://dx.doi.org/10.1016/S1359-6454(99)00412-7.
[11] A. van de Walle, G. Ceder, Automating first-principles phase diagram

calculations, J. Phase Equilib. 23 (4) (2002) 348, http://dx.doi.org/10.1361/105497102770331596.
[12] G. Ghosh, A. van de Walle, M. Asta, First-principles calculations of the structural and thermodynamic properties of bcc, fcc and hcp solid solutions in the Al-TM (TM=Ti, Zr and Hf) systems: a comparison of cluster expansion and supercell methods, Acta Mater. 56 (13) (2008) 3202–3221, http://dx.doi.org/10.1016/j.actamat.2008.03.006.
[13] B.P. Burton, A. van de Walle, First principles phase diagram calculations for the octahedral-interstitial system αTiO_X, $0 \leq X \leq 1/2$, Calphad 39 (2012) 97–103, http://dx.doi.org/10.1016/j.calphad.2012.09.004.
[14] S. Allen, J. Cahn, Ground state structures in ordered binary alloys with second neighbor interactions, Acta Metall. 20 (3) (1972) 423–433, http://dx.doi.org/10.1016/0001-6160(72)90037-5.
[15] F. Ducastelle, Order and phase stability in alloys, in: K. Terakura, H. Akai (Eds.), Interatomic Potential and Structural Stability, Springer, 1993, pp. 133–142, http://dx.doi.org/10.1007/978-3-642-84968-8_14.
[16] M. Kaburagi, J. Kanamori, A method of determining the ground state of the extended-range classical lattice gas model, Prog. Theor. Phys. 54 (1) (1975) 30–44, http://dx.doi.org/10.1143/PTP.54.30.
[17] H. Ackermann, G. Inden, R. Kikuchi, Tetrahedron approximation of the cluster variation method for b.c.c. alloys, Acta Metall. 37 (1) (1989) 1–7, http://dx.doi.org/10.1016/0001-6160(89)90259-9.
[18] G. Ceder, G.D. Garbulsky, D. Avis, K. Fukuda, Ground states of a ternary fcc lattice model with nearest- and next-nearest-neighbor interactions, Phys. Rev. B 49 (1) (1994) 1–7, http://dx.doi.org/10.1103/PhysRevB.49.1.
[19] G. Zhang, F.H. Stillinger, S. Torquato, The perfect glass paradigm: disordered hyperuniform glasses down to absolute zero, Sci. Rep 6 (2016) 36963, http://dx.doi.org/10.1038/srep36963.
[20] M. Hejna, P.J. Steinhardt, S. Torquato, Nearly hyperuniform network models of amorphous silicon, Phys. Rev. B 87 (24) (2013) 245204, http://dx.doi.org/10.1103/PhysRevB.87.245204.
[21] P.R. Cantwell, M. Tang, S.J. Dillon, J. Luo, G.S. Rohrer, M.P. Harmer, Grain boundary complexions, Acta Mater. 62 (2014) 1–48, http://dx.doi.org/10.1016/j.actamat.2013.07.037.
[22] W.D. Kaplan, D. Chatain, P. Wynblatt, W.C. Carter, A review of wetting versus adsorption, complexions, and related phenomena: the rosetta stone of wetting, J. Mater. Sci. 48 (17) (2013) 5681–5717, http://dx.doi.org/10.1007/s10853-013-7462-y.
[23] T. Frolov, M. Asta, Y. Mishin, Segregation-induced phase transformations in grain boundaries, Phys. Rev. B 92 (2) (2015) 020103, http://dx.doi.org/10.1103/PhysRevB.92.020103.
[24] Z. Pan, T.J. Rupert, Effect of grain boundary character on segregation-induced structural transitions, Phys. Rev. B 93 (13) (2016) 134113, http://dx.doi.org/10.1103/PhysRevB.93.134113.
[25] Z. Pan, T.J. Rupert, Formation of ordered and disordered interfacial films in immiscible metal alloys, Scr. Mater. 130 (2017) 91–95, http://dx.doi.org/10.1016/j.scriptamat.2016.11.025.
[26] Z. Pan, T.J. Rupert, Amorphous intergranular films as toughening structural features, Acta Mater. 89 (2015) 205–214, http://dx.doi.org/10.1016/j.actamat.2015.02.012.
[27] N. Zhou, Z. Yu, Y. Zhang, M.P. Harmer, J. Luo, Calculation and validation of a grain boundary complexion diagram for Bi-doped Ni, Scr. Mater. 130 (2017) 165–169, http://dx.doi.org/10.1016/j.scriptamat.2016.11.036.
[28] A. Kundu, K.M. Asl, J. Luo, M.P. Harmer, Identification of a bilayer grain boundary complexion in Bi-doped Cu, Scr. Mater. 68 (2) (2013) 146–149, http://dx.doi.org/10.1016/j.scriptamat.2012.10.012.
[29] J. Rickman, H. Chan, M. Harmer, J. Luo, Grain-boundary layering transitions in a model bicrystal, Surf. Sci. 618 (2013) 88–93, http://dx.doi.org/10.1016/j.susc.2013.09.004.
[30] P.R. Cantwell, S. Ma, S.A. Bojarski, G.S. Rohrer, M.P. Harmer, Expanding timetemperature-transformation (TTT) diagrams to interfaces: a new approach for grain boundary engineering, Acta Mater. 106 (2016) 78–86, http://dx.doi.org/10.1016/j.actamat.2016.01.010.
[31] W. Huang, D.A. Kitchaev, S.T. Dacek, Z. Rong, A. Urban, S. Cao, C. Luo, G. Ceder, Finding and proving the exact ground state of a generalized ising model by convex optimization and MAX-SAT, Phys. Rev. B 94 (13) (2016) 134424, http://dx.doi.org/10.1103/PhysRevB.94.134424.
[32] E. Boros, P. Hammer, Pseudo-Boolean optimization, Discret. Appl. Math. 123 (1) (2002) 155–225, http://dx.doi.org/10.1016/S0166-218X(01)00341-9.
[33] T.D. Lee, C.N. Yang, Statistical theory of equations of state and phase transitions. II. Lattice gas and Ising model, Phys. Rev. 87 (3) (1952) 410–419, http://dx.doi.org/10.1103/PhysRev.87.410.
[34] B. Grünbaum, V. Klee, M.A. Perles, G.C. Shephard, Convex Polytopes, vol. 16, Springer, 1967, http://dx.doi.org/10.1007/978-1-4613-0019-9.
[35] R.T. Rockafellar, Convex Analysis, Princeton University Press, 1970, http://dx.doi.org/10.1515/9781400873173.
[36] G.G. Brown, R.F. Dell, Formulating integer linear programs: a rogues' gallery, INFORMS Trans. Educ 7 (2) (2007) 153–159, http://dx.doi.org/10.1287/ited.7.2.153.
[37] J. Marques-Silva, Practical applications of Boolean satisfiability, in: 9th International Workshop on Discrete Event Systems, IEEE, 2008, pp. 74–80, http://dx.doi.org/10.1109/WODES.2008.4605925.
[38] G.L. Nemhauser, L.A. Wolsey, Integer Programming and Combinatorial Optimization, John Wiley & Sons, 1988.
[39] D.-S. Chen, R.G. Batson, Y. Dang, Applied Integer Programming: modeling and Solution, John Wiley & Sons, 2010, http://dx.doi.org/10.1002/9781118166000.
[40] Gurobi Optimization, Inc., Gurobi Optimizer Reference Manual, 2016. URL, http://www.gurobi.com.
[41] T. Chookajorn, C.A. Schuh, Thermodynamics of stable nanocrystalline alloys: a Monte Carlo analysis, Phys. Rev. B 89 (6) (2014) 064102, http://dx.doi.org/10.1103/PhysRevB.89.064102.
[42] A.R. Kalidindi, C.A. Schuh, A compound unit method for incorporating ordered compounds into lattice models of alloys, Comput. Mater. Sci. 118 (2016) 172–179, http://dx.doi.org/10.1016/j.commatsci.2016.02.039.
[43] A.R. Kalidindi, C.A. Schuh, Stability criteria for nanocrystalline alloys, Acta Mater. 132 (2017) 128–137, http://dx.doi.org/10.1016/j.actamat.2017.03.029.
[44] A.R. Kalidindi, C.A. Schuh, Phase transitions in stable nanocrystalline alloys, J. Mater. Res. 32 (11) (2017) 19932002, http://dx.doi.org/10.1557/jmr.2017.188.
[45] P.M. Larsen, A.R. Kalidindi, S. Schmidt, C.A. Schuh, Example Code Repository, 2017. URL, http://github.com/pmla/alloy-design.

Please cite this article in press as: P.M. Larsen, et al., Alloy design as an inverse problem of cluster expansion models, Acta Materialia (2017), http://dx.doi.org/10.1016/j.actamat.2017.08.008

Paper 6

Improving the Performance of Forward Modelling Based Pattern Indexing by Optimal Selection of Orientations

Larsen P M and Schmidt S

Submitted 12th July 2017 to *Journal of Applied Crystallography*
Revised 22nd August 2017
Awaiting decision

Improving the Performance of Forward Modelling Based Pattern Indexing by Optimal Selection of Orientations

Peter Mahler Larsen* and Søren Schmidt

Department of Physics, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark. Correspondence e-mail: pmla@fysik.dtu.dk

Physics-based forward modelling has been shown to greatly improve the robustness of pattern indexing in a range of modalities. The improvement in robustness, though, comes at the cost of a significantly increased running time. Since the running time is dependent on the number of orientations in the dictionary set, the choice of how to select orientations is of critical importance. Here, a rigorous quantification of the error of an orientation set is presented, and it is shown that this measure is equivalent to the covering radius in $\mathbb{S}^3$. Orientation sets are generated which are nearly optimal with respect to this error. In doing so, the number of orientations required to achieve a desired error is reduced, or, alternatively, the error distribution at a fixed number of orientations is reduced. The orientation sets are made available online.

1. Introduction

In many types of diffraction experiments, the aim is to determine the orientation of the diffracted volume which creates the experimentally observed pattern. For example, when studying a multigrain sample with the 3DXRD technique (Poulsen *et al.*, 2001), a 'grain map' is constructured by finding the crystalline orientation at each point in the sample. The process of determining the crystalline orientation from a diffraction pattern on the detector is known as *pattern indexing*.

The most widely-used pattern indexing methods work 'backwards' from features in the observed data to an orientation. Such methods are typically highly efficient, but can fail in the presence of noise. A well-known example is in Electron Backscatter Diffraction (EBSD), where the Hough transform is used to find lines in the backscattered Kikuchi pattern, from which the orientation can be determined (Adams *et al.*, 1993). Under noisy conditions, however, the Kichuchi lines can no longer be reliably identified and the indexing process fails as a consequence.

The desire to analyze diffraction patterns under less-than-ideal conditions has motivated the development of forward modelling based pattern indexing, also known as dictionary-based indexing. In a forward model, rather than working backwards from the data, the orientation is found using a brute-force approach. A dictionary is constructed by selecting a set of orientations, and generating *simulated* patterns for each of them. The experimental pattern is then compared against every simulated pattern in the dictionary, and the dictionary pattern with the highest similarity determines the orientation. Here, the similarity is a distance function of the simulated and experimentally observed intensities. By using the full image information instead of looking for specific features, this approach is highly robust to noise.

A significant drawback of the forward modelling approach, however, is the computational effort required: each experimental pattern must be tested against every dictionary pattern. Since, the accuracy of the pattern indexing process depends on the granularity of the dictionary, a more accurate indexing requires a larger dictionary. As the running time also depends on the size of the dictionary, we ask the question: how can we achieve the highest accuracy with the fewest dictionary orientations? In this article, we describe a method for doing so with the use of quaternions.

1.1. Measurement of Dictionary Orientation Sets

Orientations can be conveniently represented using unit quaternions (Altmann, 2005). Briefly, a quaternion is a four dimensional vector of the form $q = \{w, ix, jy, kz\}$, where w, x, y and z are real numbers, and i, j and k are imaginary numbers which generalize the better-known complex numbers. Unit quaternions represent points on a four-dimensional hypersphere, a space formally known as $\mathbb{S}^3$ and which consists of all vectors which satisfy $\sqrt{w^2+x^2+y^2+z^2} = 1$. This space is a double covering of $SO(3)$, the group of rotations in three-dimensional Euclidean space $\left(\mathbb{R}^3\right)$. The double covering relationship means that $-q$ and q represent the same orientation, which is evident when considering the quaternion-derived rotation matrix:

$$\mathbf{U}_q = \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2wz & 2xz+2wy \\ 2xy+2wz & 1-2x^2-2z^2 & 2yz-2wx \\ 2xz-2wy & 2yz+2wx & 1-2x^2-2y^2 \end{bmatrix} \quad (1)$$

It can be seen that in each element of $\mathbf{U}_q$, the sign of the quaternion cancels out. By using the quaternion representation, the problem of selecting an optimal set of dictionary orientations is equivalent to finding an optimal distribution of a set of points on $\mathbb{S}^3$. To do so, we must first decide what constitutes a good distribution.

The misorientation between two orientations in quaternion form, p and q, is given by:

$$\alpha\left(p,q\right)=2\arccos\left|\langle p,q\rangle\right| \tag{2}$$

where $\langle p,q\rangle$ denotes the inner product of p and q. In many previous studies, dictionary orientation sets are quantified by the misorientation between neighbouring orientations, for example, the average value of $\alpha\left(p,q\right)$ over all pairs of nearest neighbours p and q. This may be adequate when the orientation set has a known, grid-like structure, but it does not constitute a universal measure of quality. To illustrate this with a pathological example, consider an orientation set, $\mathbf{Q}$, where all orientations lie at the same point. The misorientation between all pairs of orientations is zero, that is

$$\alpha(p,q)=0\quad\forall p\in\mathbf{Q},q\in\mathbf{Q} \tag{3}$$

yet the set constitutes the worst possible dictionary. A good measure of quality should instead consider the misorientation between the dictionary set and any possible experimental orientation. We define the error term as the maximum misorientation between these two, i.e. how far can an experimental orientation lie from the dictionary? More specifically, this error term is given by:

$$\alpha_{\max}=\max_{x\in SO(3)}\min_{q\in\mathbf{Q}}\alpha\left(x,q\right) \tag{4}$$

This quantity can be minimized by solving the *spherical covering problem* in $\mathbb{S}^3$. Given N hyperspherical caps of equal radius, r, called the *covering radius*, the spherical covering problem asks how to arrange the caps to cover the surface of $\mathbb{S}^3$ with minimal r. We describe this problem in detail in Section 2.

By creating orientation sets with a small covering radius, we can either reduce the number of orientations required to achieve a desired error tolerance (thereby reducing the running time of forward modelling pattern indexing), or simply improve the error distribution for a fixed number of orientations. Creation of such sets is the principal contribution of this work.

1.2. Previous Work

Forward modelling has been successfully applied in many types of diffraction-based experiments, including the indexing of 3D X-ray diffraction microscopy data (Li & Suter, 2013; Schmidt, 2014), EBSD data (Chen *et al.*, 2015) and electron channeling patterns (Singh & De Graef, 2017). Any forward modelling method requires a discretization of $SO(3)$. Whilst many such discretization methods have been developed, here we consider only three which are both successful and commonly used amongst crystallographers.

Yershova et al. (Yershova *et al.*, 2010) have developed an incremental infinite sequence based on the Hopf fibration. The method generates orientations deterministically, with proven maximal dispersion reduction when used as a sequence. Furthermore, the orientation sets are isolatitudinal, which permits expansion into spherical harmonics (Dahms & Bunge, 1989), refinable, and can be generated on-the-fly. Whilst the method has many desirable properties, it is developed for the purpose of robot motion planning and is not easily integrated with crystallographic fundamental zones. To remedy this, Roşca et al. (Roşca *et al.*, 2014) have developed 'cubochoric' coordinates, in which an area-preserving Lambert projection is used to map points from a cubic grid onto any desired crystallographic fundamental zone in $SO(3)$. A different approach, developed by Karney (Karney, 2007) for use in molecular modelling, is to generate sets which attempt to solve the spherical covering problem. Inspired by the observation that body-centred cubic (BCC) grids solve the covering problem in $\mathbb{R}^3$, BCC grids are constructed in Rodrigues-Frank (RF) space (Frank, 1988; Morawiec & Field, 1996) in order to generate good coverings in $SO(3)$.

Table 1
Summary of properties of different methods of orientation set generation. Existing methods prioritize fast generation and a grid-like structure. In our work we optimize the covering radius at the expense of all other properties. The optimality gap for a set of N orientations is the percentage difference of its covering radius to that of the simplex bound (c.f. Section 2.5). [1]Non-isolatitudinal sets do not permit an expansion into spherical harmonics, though any orientation set can be expanded into hyperspherical harmonics (Mason & Schuh, 2008; Mason, 2009). [2]These orientation sets can be mapped out into 7 of 11 Laue group fundamental zones (c.f. Section 2.4).

Method	Fast generation	Refinable	Isolatitudinal	Isochoric	Crystallographic	Opt. gap at $N\approx10^5$
Random sampling	✓	✓	-[1]	-	✓	127%
Hopf fibration	✓	✓	✓	-	-	59.9%
Cubochoric	✓	✓	✓	✓	✓	40.8%
Octahedral BCC	✓	✓	-[1]	-	-[2]	15.4%
Present work	-	-	-[1]	-	✓	4.64%

Table 1 summarizes the properties of the different methods of generating orientation sets. Each of the three aforementioned techniques attempts to solve slightly different problems and involves different trade-offs as a consequence, although one feature they have in common is fast generation. We take an alternative approach, sacrificing other properties in pursuit of creating the 'best' possible orientation sets. Whilst this approach requires a significant up-front computational effort, this is a good trade-off when the resulting sets will subsequently be used many times. We emphasize that whilst the orientational error is critical to forward modelling, there are many other sources of error in any modality (see Ram et al. (Ram *et al.*, 2017) for a comprehensive analysis in an EBSD context).

The rest of this article is organized as follows: in Section 2 we define the spherical covering problem on $\mathbb{S}^d$, show how this relates to the problem of finding an optimal set of orientations, and derive a conjectured lower bound. We describe the generation of orientation sets in Section 3. Results on the covering radius and error distributions of the resulting orientation are given in Section 4. Lastly, the advantages and drawbacks of the method presented are discussed in Section 5.

2. Error Quantification of Orientation Sets

In order to compare different orientations sets we must define a measure of quality. Here, we describe the covering radius of a

set, which we argue is the canonical error measure since it determines the maximum possible error. We will first describe the sphere covering problem for Euclidean and spherical geometries, and then show that the problem of generating optimal orientation sets is a special case of the spherical covering problem.

2.1. Spherical Coverings

The sphere covering problem is best known in Euclidean geometries. In $\mathbb{R}^d$, it asks 'for the most economical way to cover d-dimensional space with equal overlapping spheres' (Conway & Sloane, 1998). Optimal coverings are known for $d = 1$ and $d = 2$, which are equally spaced points on a line and a hexagonal lattice, respectively, and optimal *lattice coverings* are known for $1 \leq d \leq 5$.

The presence of curvature in spherical geometries renders the covering problem vastly more challenging. In $\mathbb{S}^d$, the spherical covering problem asks for the most economical way to cover the surface of $\mathbb{S}^d$ with equal overlapping hyperspherical caps. In $\mathbb{S}^1$, the optimal covering is a set of N points with angle $\frac{2\pi}{N}$ between adjacent points. For $d > 1$, however, there is no general formula for determining the optimal spherical covering. Furthermore, unlike in $\mathbb{R}^d$, the configuration of the optimal covering depends on the number of points in the covering. For example, for $d = 2$, the known optimal configurations are the vertices of the tetrahedron, the octahedron and the icosahedron. Hardin et al. have found *putatively* optimal coverings (Hardin *et al.*, 2017) for $d = 2$ at other values of N, but these have been found using numerical optimization and are not provably optimal.

2.2. Covering Radius and Covering Density

For coverings on $\mathbb{S}^d$, the two (equivalent) measures of quality are the covering radius and the covering density. Given a discrete collection of points $\mathbf{P} = \{p_1, p_2, \ldots, p_N\} \in \mathbb{S}^d$, the covering radius, θ, is defined as the largest angular distance between any point in $\mathbb{S}^d$ and $\mathbf{P}$, that is

$$\theta = \max_{x \in \mathbb{S}^d} \min_{p \in \mathbf{P}} \arccos\langle x, p \rangle \tag{5}$$

where $\langle x, p \rangle$ denotes the inner product of x and p. Then, $\mathbf{P}$ covers the surface of $\mathbb{S}^d$ with $N = |\mathbf{P}|$ equal hyperspherical caps of radius θ. The covering density, $\tau_d(\theta)$, is given by ratio of the sum of the surface area of the caps to the surface area of unit d-sphere,

$$\tau_d(\theta) = N \frac{C_d(\theta)}{S_d(1)} \tag{6}$$

where

$$C_d(\theta) = \int_0^{\tan(\theta)} \frac{S_{d-1}(r)}{(1+r^2)^2} dr, \quad S_{d-1}(\theta) = \frac{d\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)} \theta^{d-1} \tag{7}$$

where $S_{d-1}(\theta)$ is the surface area of the d-sphere of radius θ and $C_d(\theta)$ is the surface area of a hyperspherical cap of radius θ (c.f. Appendix A for derivation). To find the covering radius, we need to determine the Voronoi cell of each point $p_i \in \mathbf{P}$. The Voronoi cell of point p_i, denoted $\mathrm{Vor}(p_i)$, consists of all points of $\mathbb{S}^d$ that are at least as close to p_i as to any other p_j. More specifically:

$$\mathrm{Vor}(p_i) = \{x \in \mathbb{S}^d \mid \arccos\langle x, p_i \rangle \leq \arccos\langle x, p_j \rangle \ \ \forall j\} \tag{8}$$

Since the vertices of the Voronoi cells are the points which locally maximize the angular distance from $\mathbf{P}$, the covering radius is determined by the Voronoi vertex that lies furthest from $\mathbf{P}$.

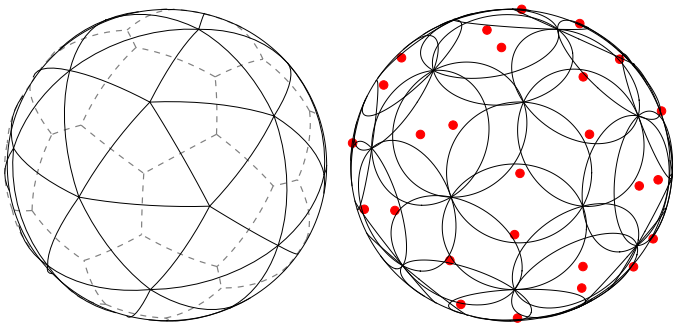


Figure 1
Left: a putatively optimal spherical covering for 28 points in $\mathbb{S}^2$ (point set due to Hardin et al. (Hardin *et al.*, 2017)). The solid lines indicate the spherically constrained Delaunay triangulation. The dashed lines indicate the Voronoi cells. **Right:** the same points; each simplex in the Delaunay triangulation has a circumcap, the centre of which (marked in red) lies at a Voronoi cell vertex. The maximum simplex circumradius determines the covering radius of the point set.

The Voronoi cells of a set of points in $\mathbb{S}^d$ are not easy to calculate directly, so instead we calculate the (hyperspherically constrained) Delaunay triangulation. The Delaunay triangulation, $DT(\mathbf{P})$, is a set of hyperspherical simplices whose vertices are points in $\mathbf{P}$ which satisfy the empty-sphere condition, that is, no points in $\mathbf{P}$ lie inside the circumhypercap of any simplex in $DT(\mathbf{P})$. Associated with each simplex is a Voronoi vertex, which lies at the centre of the simplex circumhypercap (the circumcentre). The Delaunay triangulation, Voronoi cells and simplex circumhypercaps and circumcentres are illustrated in $\mathbb{S}^2$ in Figure 1. We now show how to calculate the circumcentre of a simplex.

Theorem. *For a hyperspherical simplex* $\mathbf{t} \in DT(\mathbf{P})$ *with vertices* $\{p_1, p_2, p_3, \ldots, p_{d+1}\} \in \mathbb{S}^d$*, the position of the circumcentre, X, is equal to the unit normal vector of the d-dimensional hyperplane on which the vertices of* $\mathbf{t}$ *lie.*

Proof. Let $\mathbf{S} = \{s_1 = p_2 - p_1, s_2 = p_3 - p_1, s_3 = p_4 - p_1, \ldots, s_d = p_{d+1} - p_1\}$ and let $X \in \mathbb{S}^d$ be the circumcentre of $\mathbf{t}$. Then, per definition, X must satisfy:

$$p_i \cdot X = p_1 \cdot X \quad \forall i \tag{9}$$

Subtracting $p_1 \cdot X$ from each side gives:

$$s_i \cdot X = 0 \quad \forall i \tag{10}$$

The unit length of X follows from requiring $X \in \mathbb{S}^d$. □

To find X, we calculate the normalized d-fold vector cross product (Brown & Gray, 1967) of $\mathbf{S}$. Since every hyperplane has two (opposite) unit plane normals, X has two solutions, which correspond to the centre of the simplex hypercircumcap and its dual. However, given that $|\mathbf{P}| \geq d+2$ only one of these solutions fulfils the empty-sphere condition, which is the one which satisfies: $\langle X, p_i \rangle > 0 \ \ \forall i$. This corresponds to the smaller of the two hypercircumcaps. For a set of points on $\mathbb{S}^d$, the vertices of

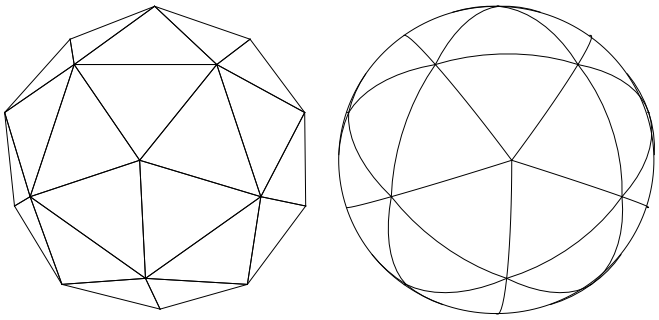


Figure 2
Convex hull (left) and the spherically constrained Delaunay triangulation (right) of 22 points on the sphere. The triangulations exist in $\mathbb{R}^3$ and $\mathbb{S}^2$ respectively, but the vertices of each simplex are the same. Data due to Hardin et al. (Hardin *et al.*, 2017).

each simplex $\mathbf{t} \in DT(\mathbf{P})$ can be found by calculating the convex hull of $\mathbf{P}$, as shown in Figure 2. If we denote the circumradius of a simplex $\mathbf{t}$ by $\phi(\mathbf{t})$, Equation (5) can be restated as:

$$\theta = \max\{\phi(\mathbf{t}) \mid \mathbf{t} \in DT(\mathbf{P})\} \tag{11}$$

which provides a practical solution to Equation (5): the covering radius of a point set is simply the maximum simplex circumradius.

2.3. Orientation Sets

The problem of finding a good spherical covering is immediately relatable to the problem of finding good sets of orientations. As described in the introduction, rotations can be represented by quaternions, which are points on $\mathbb{S}^3$. The maximum rotational angle between a point $x \in SO(3)$ and a point set $\mathbf{P}$, also called the maximum *misorientation*, is given by:

$$\begin{aligned} \alpha_{\max} &= 2 \max_{x \in \mathbb{S}^3} \min_{p \in \mathbf{P}} \min \left[\arccos\langle -x, p \rangle, \arccos\langle x, p \rangle \right] \\ &= 2 \max_{x \in \mathbb{S}^3} \min_{p \in \mathbf{Q}} \arccos\langle x, p \rangle \end{aligned} \tag{12}$$

where $\mathbf{Q} = \mathbf{P} \cup \{-p \mid p \in \mathbf{P}\}$. It can be seen that, for a point set with antipodal symmetry, $\alpha_{\max} = 2\theta$, that is, the maximum misorientation is twice the covering radius. Thus, the problem of finding a set of rotations with the lowest maximum misorientation is equivalent to finding an optimal spherical covering for a point set with antipodal symmetry on $\mathbb{S}^3$.

2.4. Integration with Crystallographic Symmetries

Equation (12) shows that a set of $2N$ points with antipodal symmetry represents a set of N rotations. A set of orientations generated in this way covers the whole space of $SO(3)$, and is immediately applicable to pattern indexing of materials with triclinic (C_1) Bravais lattices. For materials with higher order symmetry, though, a dictionary set which covers all of $SO(3)$ is wasteful, since only the fundamental zone orientations (He & Jonas, 2007) are needed. A naive approach for selecting fundamental zone orientations is to generate a full covering of $SO(3)$ and then simply 'cut out' the desired region; this introduces artifacts at the boundaries of the fundamental zone which increase the covering radius significantly. Instead, we apply the symmetry of the desired point group during generation of the orientation sets.

Given a set of basis points $\mathbf{B} = \{b_1, b_2, \ldots\}$ and a quaternion group $\mathbf{G} = \{g_1, g_2, \ldots\}$, we can create a set of points with the symmetry of $\mathbf{G}$ by:

$$\mathbf{P} = \{b \otimes g \mid b \in \mathbf{B}, g \in \mathbf{G}\} \tag{13}$$

where $\otimes$ denotes quaternion multiplication. If $\mathbf{P}$ is to represent a set of orientations (c.f. Equation (12)), $\mathbf{G}$ must be a superset of antipodal symmetry (C_1). The finite quaternion groups which meet this requirement are (Conway & Smith, 2003):

$2I_{60}$	The binary icosahedral group
$2O_{24}$	The binary octahedral group
$2T_{12}$	The binary tetrahedral group
$2D_n$	The binary dihedral group
$2C_n$	The binary cyclic group

With the exception of the binary icosahedral group, each of these is used to describe the generators of the 11 Laue groups (Morawiec, 2003), $C_1, C_2, C_3, C_4, C_6, D_2, D_3, D_4, D_6, T$ and O. By the application of a symmetry group, the problem of finding a good spherical covering for a chosen crystallographic fundamental zone is reduced to a problem of finding an optimal configuration of the basis points. The Laue groups can be divided into two sets:

$$\{C_2, C_4, D_2, D_4, T, O\} \tag{14}$$

and

$$\{C_3, C_6, D_3, D_6\} \tag{15}$$

where the elements of each are subsets of O and D_6 respectively (C_1 is trivially a subset of both). This means that, if we generate sphere coverings with O and D_6 applied according to Equation (13), then by an appropriate mapping of the fundamental zone orientations we obtain sphere coverings for *all* Laue groups, without the aforementioned boundary artifacts. The Laue group subset relationships are shown in Appendix B.

2.5. Derivation of the Simplex Bound on $\mathbb{S}^3$

In addition to knowing the covering radius and density of a point set, it is useful to know how far from optimality a set is. We can estimate the optimality gap with a lower bound. The simplex bound is a classic result which gives an upper bound

on the density of sphere packings, and a lower bound on the density of sphere coverings. It has been proven for packings in $\mathbb{R}^d$ (Rogers, 1958) and $\mathbb{S}^d$ (Böröczky, K, 1978), and for coverings in $\mathbb{R}^d$ (Coxeter *et al.*, 1959) and $\mathbb{S}^2$ (Tóth, 1964). Böröczky has conjectured that it is a lower bound on $\mathbb{S}^3$ (Böröczky, 2004). Despite lacking a proof, we will use the simplex bound on $\mathbb{S}^3$ to estimate the optimality of our point sets, as it is 'intuitively obvious'.

The premise of the simplex covering bound is that the lowest covering density can be achieved with regular simplices. This concept is illustrated in Figure 3. Regular simplices tesselate in $\mathbb{R}^1$ and $\mathbb{R}^2$. In $\mathbb{R}^d$ for $d \geq 3$ regular simplices do not tesselate, and thus the simplex covering density is an unattainable lower bound. As stated previously, regular simplices tesselate in $\mathbb{S}^2$ for three configurations: the tetrahedron, the octahedron and the icosahedron. Thus, the simplex bound is tight for these configurations only, and is provably unattainable for any other number of vertices. In $\mathbb{S}^3$, regular simplices tesselate only in the 5-cell, the 16-cell and the 600-cell. If Böröczky's conjecture is

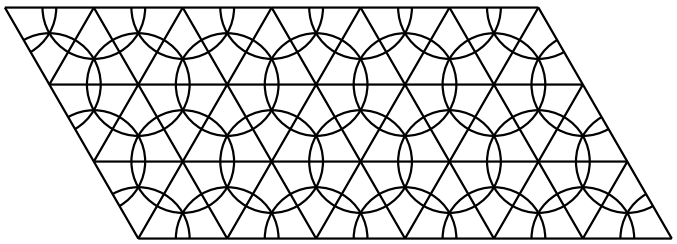


Figure 3
Illustration of the simplex bound in $\mathbb{R}^2$, shown here due to the difficulty of visualizing the simplex bound in $\mathbb{S}^3$. Regular simplices in $\mathbb{R}^2$ are equilateral triangles, which tessellate. At the vertices of each triangle (of circumradius r) is a circle of radius r. The area of intersection between a triangle and a circle is a circular sector of angle $\frac{\pi}{3}$. Each triangle is covered by three equal areas of intersection. The covering density is therefore the ratio of the sum of the three areas of intersection to the area of the triangle: $\tau_{\mathbb{R}^2} = \frac{2\pi}{3\sqrt{3}}$. In $\mathbb{R}^d$ the covering density is independent of r, which is not the case in $\mathbb{S}^d$ for $d \geq 2$ due to a lack of tesselation.

correct, the simplex bound is tight only for these configurations. Since no description of the simplex bound covering density on $\mathbb{S}^3$ could be found in the literature, we derive an expression for it here.

Given a hyperspherical cap on $\mathbb{S}^3$ of radius θ and volume $C_3(\theta)$, we denote the inscribed regular spherical tetrahedron $T(\theta)$. At each of the four vertices of $T(\theta)$ is a hyperspherical cap of radius θ. Each of these caps intersects $T(\theta)$ with solid angle $\Omega(\theta)$, giving a volume of intersection of $C_3(\theta)\frac{\Omega(\theta)}{4\pi}$. Now $T(\theta)$ is covered by the four equal volumes of intersection. The covering density, $\tau_{\mathbb{S}^3}$, is the ratio of the sum of the four volumes of intersection to the volume of $T(\theta)$:

$$\tau_{\mathbb{S}^3}(\theta) = 4\, C_3(\theta)\, \frac{\Omega(\theta)}{4\pi} \frac{1}{\mathrm{Vol}(T(\theta))} \tag{16}$$

where:

$$C_3(\theta) = \pi(2\theta - \sin(2\theta)) \tag{17}$$

$$\Omega(\theta) = 3\psi(\theta) - \pi \tag{18}$$

$$\psi(\theta) = \arccos\left(\frac{4\cos^2(\theta) - 1}{8\cos^2(\theta) + 1}\right) \tag{19}$$

$$\mathrm{Vol}(T(\theta)) = \Big(-\mathrm{Re}(L) + \pi(\arg(-Q) + 3\psi(\theta)) - \frac{3}{2}\pi^2\Big) \mod 2\pi^2 \tag{20}$$

$$Q = 3e^{-2i\psi(\theta)} + 4e^{-3i\psi(\theta)} + e^{-6i\psi(\theta)} \tag{21}$$

$$L = \frac{1}{2}\Big[\mathrm{Li}_2\left(Z_0\right) + 3\,\mathrm{Li}_2\left(Z_0 e^{-4i\psi(\theta)}\right) - 4\,\mathrm{Li}_2\left(-Z_0 e^{-3i\psi(\theta)}\right) - 3\psi(\theta)^2\Big] \tag{22}$$

$$Z_0 = \frac{-6\sin^2(\psi(\theta))}{Q} + \frac{2\sqrt{(\cos(\psi(\theta))+1)^3(1-3\cos(\psi(\theta)))}}{Q} \tag{23}$$

where $\psi(\theta)$ is the dihedral angle of $T(\theta)$. The terms in Equations (17) - (19) are derived in A. Equations (20) - (23) are a simplification of Murakami's formula for the volume of a spherical tetrahedron (Murakami, 2012), for the case where all six dihedral angles are equal (a regular spherical tetrahedron).

The covering density can be used to estimate the optimality gap of a point set. For a set of N points with covering radius θ, the lower bound on the covering radius θ^* can be found by rearranging the density expression in Equation (6):

$$N = \frac{2\pi^2 \tau_{\mathbb{S}^3}(\theta^*)}{C_3(\theta^*)} \tag{24}$$

where $2\pi^2$ is the surface area of $\mathbb{S}^3$. The optimality gap of the point set is then $\theta/\theta^* - 1$. Since $\tau_{\mathbb{S}^3}(\theta^*)$ is a nontrivial expression, we find θ^* numerically.

3. Method of Orientation Set Generation

We now describe the method for generating point sets with small covering radii. The direct problem formulation with the application of symmetry is shown in Model 1. This is essentially just a restatement of Equations (11) and (13).

Variables:	$\mathbf{B} = \{b_1 \in \mathbb{S}^3, b_2 \in \mathbb{S}^3 \ldots\}$
Parameters:	$\mathbf{G} = \{g_1 \in \mathbb{S}^3, g_2 \in \mathbb{S}^3, \ldots\}$
Minimize:	$\theta = \max\{\phi(\mathbf{t}) \mid \mathbf{t} \in DT(\mathbf{P})\}$
Subject to:	$\mathbf{P} = \{b \otimes g \mid b \in \mathbf{B}, g \in \mathbf{G}\}$

Model 1
Direct model for minimizing the covering radius of a point set in $\mathbb{S}^3$. The point set $\mathbf{P}$ is composed of a basis set, $\mathbf{B}$, on which a chosen symmetry group, $\mathbf{G}$, acts. The covering radius, θ, is calculated using the Delaunay triangulation of $\mathbf{P}$.

The problem of finding optimal spherical coverings is difficult; in addition to being a NP-hard problem (van Emde Boas, 1981), the objective function is non-differentiable, and the 'fitness landscape' is non-convex and has many local minima. One possible solution approach (used by Hardin et al. (Hardin *et al.*, 2017) to generate coverings in $\mathbb{S}^2$) is to use direct search. This overcomes the non-differentiability of the objective function, but repeated solution from many different starting configurations is required to find the globally optimal configuration. Furthermore, due to the poor scaling of direct search methods scale with increasing problem size, this approach is not practical since we wish to create very large orientation sets.

Since it is unlikely that we will find globally optimal solutions for large point sets with direct search, we will instead attempt to find good solutions with an indirect method. We proceed as follows: an initial set of orientations is created by sampling randomly from a uniform distribution on $SO(3)$ (Shoemake, 1992). The covering radius is then successively reduced, firstly by using gradient descent to find a configuration which is a local minimizer of the Riesz energy. Secondly, a smoothing procedure is used to improve the characteristics of the Delaunay triangulation. Lastly, a local optimization procedure is used to further refine the solution. We present no theoretical basis for the choice of methods, nor for the order in which the methods are applied. Rather, empirical experimentation has shown that the method is effective and produces point sets with a small covering radius.

The motivation for choosing these methods is illustrated in Figure 4. The random point set has a large covering radius. By minimizing the Riesz energy the covering radius is significantly reduced. The covering radius can be further reduced as shown in the optimal covering. The effect of the smoothing procedure is not shown here, as it is visually very similar to the Riesz energy and optimal covering configurations. In the rest of this Section we describe each method in detail.

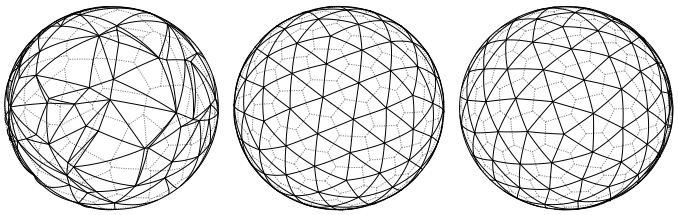


Figure 4
Delaunay triangulations and Voronoi cells of three different point sets in $\mathbb{S}^2$, for $N = 130$. **Left:** Points sampled uniformly from a random distribution. **Centre:** The global minimum configuration for the Riesz energy, here for $s = 1$ (point set due to Wales et al. (Wales & Ulker, 2006)). **Right:** Putatively optimal spherical-covering configuration (point set due to Hardin et al. (Hardin *et al.*, 2017)). Point sets in $\mathbb{S}^2$ are used here for illustrative purposes only, due to the difficulty of visualizing $\mathbb{S}^3$.

3.1. Riesz Energy Minimization

For a set of points $\mathbf{P} = \{p_1, p_2, \ldots, p_N\} \in \mathbb{S}^d$, the Riesz energy is defined as:

$$E_s(\mathbf{P}) = \begin{cases} \sum_{i \neq j}^{N} \frac{1}{|p_i - p_j|^s} & \text{if } s > 0 \\ \sum_{i \neq j}^{N} \log \frac{1}{|p_i - p_j|} & \text{if } s = 0 \end{cases} \tag{25}$$

The problem of finding optimal Riesz energy configurations is well studied, most commonly for $(d = 3, s = 1)$ (also known as the Thomson problem) (Erber & Hockney, 1991; Altschuler *et al.*, 1994; Wales & Ulker, 2006), but also for $(d = 4, s = 1)$ (Altschuler & Perez-Garrido, 2007), and in the general case (Hardin & Saff, 2004; Rakhmanov *et al.*, 1995). The sphere-packing problem is equivalent to solving for $s = \infty$. Cohn et al. (Cohn & Kumar, 2007) have shown that there exist configurations for certain values of N which are universally optimal, that is, globally optimal solutions for every value of s. The known universally optimal configurations for $d = 3$ are the tetrahedron, the 16-cell and the 600-cell. The vertices of these polyhedra are conjectured to be global optima for the sphere-covering problem, since their Delaunay triangulations consist of regular spherical tetrahedra (c.f. Section 2.5). However, for any value of N for which a universally optimal configuration does not exist, there is no value of s which for a configuration minimizing $E_s(\mathbf{P})$ guarantees an optimal spherical covering. As such, we will select a value of s on the following basis: Kuijlaars et al. (Kuijlaars *et al.*, 2007) have shown that the set of points $\mathbf{P}$ which minimizes $E_s(\mathbf{P})$ is well-distributed when $d - 1 \leq s < d$. We will select $s = 2$ since longer range potentials exhibit fewer local minima (Wales & Ulker, 2006). We have used the PR+ conjugate gradient method (Wright & Nocedal, 1999) to find a local minimum of $E_s(\mathbf{P})$. The resulting configuration is a good intermediate solution with a small covering radius.

3.2. Optimal Delaunay Triangulation Smoothing

Minimizing the Riesz energy of a point set reduces the covering radius whilst considering only the relative positions of the points. We can obtain a further reduction in covering radius by considering the positions of a point set *and* the simplices in its Delaunay triangulation. This is a well-studied problem in the computational geometry community known as *tetrahedral meshing*. Given a set of points sampled from an object (e.g. a teapot model) the objective is to move the points in order to create a 'nice' Delaunay triangulation (the mesh) whilst preserving the shape of the object. Chen et al. (Chen, 2004) define an optimal Delaunay triangulation as a set of points which minimizes the energy function:

$$E_{\text{ODT}} = \frac{1}{d+1} \sum_{i=1 \ldots N} \int_{\Omega_i} ||p - p_i||^2 dp \tag{26}$$

where Ω_i is the 1-ring of p_i (the volume bounded by p_i and its simplicial neighbours). Minimization of this energy results in a Delaunay triangulation whose simplices have a low circumradius to inradius ratio. Alliez et al. (Alliez *et al.*, 2005) have

shown that, for a given point, the position which minimizes E_{ODT} is:

$$p_i^* = \frac{1}{\mathrm{Vol}\,(\Omega_i)} \sum_{\mathbf{t}\in\mathbf{\Omega}_i} \mathrm{Vol}(\mathbf{t}) C(\mathbf{t}) \tag{27}$$

where $\mathrm{Vol}(\mathbf{t})$ and $C(\mathbf{t})$ are respectively the volume and circumcentre of simplex $\mathbf{t}$. They have shown that the energy can be minimized with guaranteed convergence by alternately constructing the Delaunay triangulation, and moving the vertices to their optimal positions using Equation (27).

For our applications the 'object' whose shape we must preserve is simply $\mathbb{S}^3$. As such, after calculating the optimal vertex position using Equation (27) the vertex position is normalized in order to bring it back onto $\mathbb{S}^3$. We also calculate $\mathrm{Vol}(\mathbf{t})$ for a spherical tetrahedron (Murakami, 2012) rather than for a Euclidean tetrahedron. Despite the intended use for Euclidean geometries, we have found that this method works very well in practice in $\mathbb{S}^3$, which is likely due to the small local curvature of $\mathbb{S}^3$ for large point sets.

3.3. Local Refinement

As a last step in the process of reducing the covering radius, we use an optimization procedure to iteratively refine a succession of local neighbourhoods. We do so by generalizing the direct problem, by iteratively dividing **B** into an active set **A** and a constant set **C**. We then minimize the maximum circumradius of the simplices with a vertex in **A**. A description of the optimization problem is given in Model 2.

Whilst the smallest active set consists of a single vertex, we find that optimizing the vertices of a whole simplex at a time gives better results. To do so, we alternately construct the Delaunay triangulation, and then optimize each simplex in turn. The order in which the simplices are optimized is determined by their circumradius, from largest to smallest. After each update the chosen symmetry group is reapplied to the basis set in order to maintain a consistent point set.

Since the minimization the maximum value of a set is a non-differentiable objective function, we use the Nelder-Mead method (Nelder & Mead, 1965) to optimize the above function as it is a derivative-free method. In order to avoid dealing with the implicit constraint $|p| = 1 \;\; \forall p \in \mathbf{P}$, we represent the vertices using RF vectors. Representing the vertices as RF vectors during optimization has the added benefit of reducing the number of variables, which is particularly advantageous when using the Nelder-Mead method. Since a RF vector representation of any 180° rotation has infinite magnitude, we rotate the local neighbourhood under consideration to $\{1,0,0,0\}$ prior to optimization, and back again after optimization.

Variables:	$\mathbf{A} = \{a_1 \in \mathbb{S}^3, a_2 \in \mathbb{S}^3, \ldots\}$	(1)
Parameters:	$\mathbf{C} = \{c_1 \in \mathbb{S}^3, c_2 \in \mathbb{S}^3, \ldots\}$	(2)
	$\mathbf{G} = \{g_1 \in \mathbb{S}^3, g_2 \in \mathbb{S}^3, \ldots\}$	(3)
Minimize:	$\max\{\phi(\mathbf{t}) \mid \mathbf{t} \in DT(\mathbf{P}) \;\wedge\; \mathbf{t} \cap \mathbf{A} \neq \emptyset\}$	(4)
Subject to:	$\mathbf{B} = \mathbf{A} \cup \mathbf{C}$	(5)
	$\mathbf{P} = \{b \otimes g \mid b \in \mathbf{B}, g \in \mathbf{G}\}$	(6)

Model 2

Model for reducing the covering radius of a *local* neighbourhood of a point set. The point set **P** is composed of a basis set, **B**, on which a chosen symmetry group, **G**, acts. The basis set, **B**, consists of an active set, **A**, which defines the local neighbourhood to be optimized, and a constant set, **C**, which contains the remaining points. The covering radius, θ, is again calculated using the Delaunay triangulation, though only of the points which are either active or which share a simplicial neighbour with an active point.

4. Results

Figure 5 illustrates how each stage of the optimization process affects the solution quality. The initial random sampling results in a distribution of simplex circumradii that is approximately Gaussian. Minimization of the Riesz energy significantly reduces the mean and variance of the simplex circumradii, as well as the number of simplices. The distribution resembles a bimodal Gaussian distribution, which suggests an ordered underlying simplex structure. Application of ODT smoothing reduces the mean and variance of the of simplex circumradii, and results, again, in an approximately Gaussian distribution. Lastly, the objective of the local refinement procedure is to minimize the maximum simplex circumradius. It can be seen that this produces a peak around the maximum circumradius with a tail of smaller circumradii below this.

Figure 6 shows a comparison of our method with the methods discussed in Section 1, in the range $N = [960, 200000]$. We have applied $2I_{60}$ symmetry, as it requires a small basis set and thus allows us to quickly generate coverings of the full space of $\mathbb{S}^3$. For each value of N, we have applied our method from 200 random starting configuration and taken the point set with the lowest covering radius. It can be seen that the resulting sets have a lower covering radius than the other methods, both at small and large values of N. Furthermore, our method displays a smooth decrease in covering radius with increasing N, which is highlighted by the almost constant covering density. We do not claim optimality for any of our point sets; in most cases the

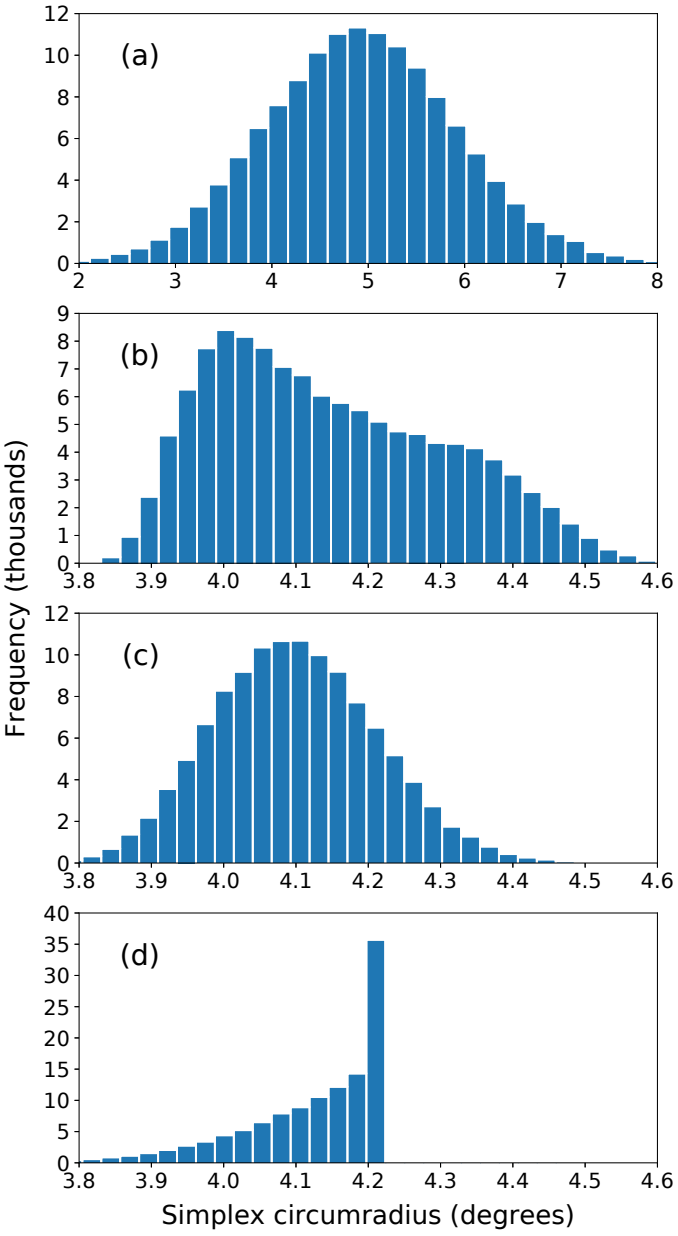


Figure 5
Histograms showing the change in simplex circumradius at each stage in the optimization for a point set with antipodal symmetry and $N = 20000$. The histograms show the simplex circumradius distribution after: (a) initial random sampling, (b) Riesz energy minimization, (c) ODT smoothing, (d) local refinement. The maximum circumradius is reduced at every stage.

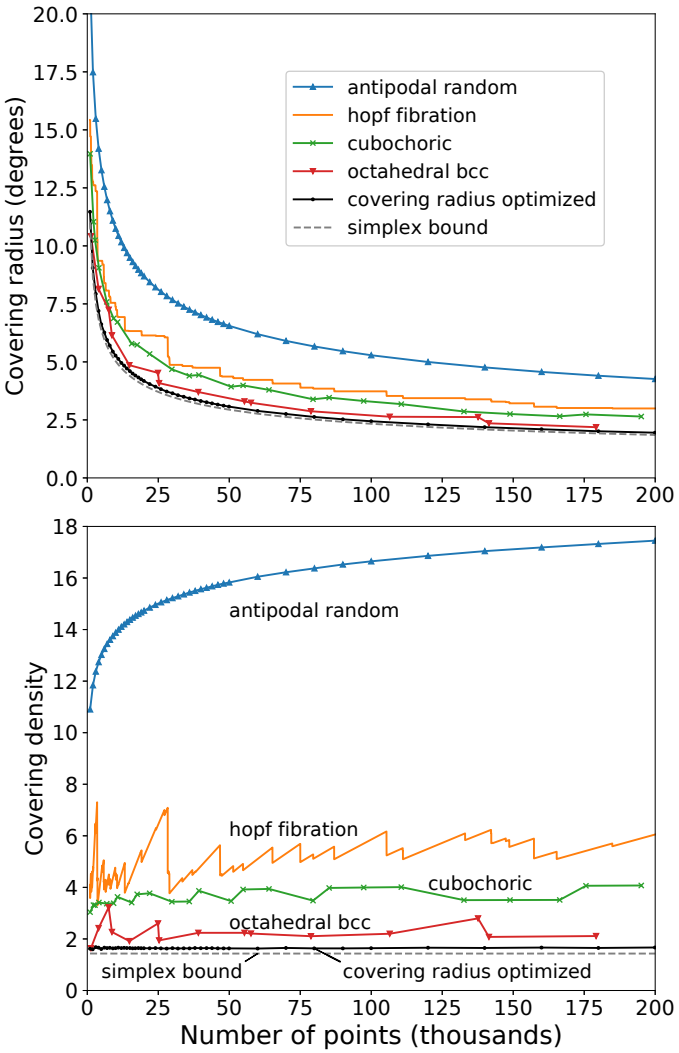


Figure 6
Comparison of the covering radius (left) and the covering density (right) of random sampling from a uniform distribution with antipodal symmetry, incremental grids based on the Hopf fibration (Yershova *et al.*, 2010), cubochoric grids (Roşca *et al.*, 2014), BCC grids with binary octahedral symmetry (Karney, 2007), and our method. For the random sampling, the mean of 10^5 runs was used. For the incremental grids based on the Hopf fibration, the covering radius was calculated at every value of N in the range shown. For the covering radius optimized point sets (our method), the best result of 200 runs was used.

covering radius of best point set was unique amongst the 200 runs. As such we can conclude that lower covering radii could be obtained simply by increasing the number of runs, though this is very time consuming for large point sets.

The optimality gaps of some selected point sets generated using our method are shown in Table 2. The gaps are below 6% at every value of N. In the Euclidean limit ($N \to \infty$) the curvature of $\mathbb{S}^3$ in a local area is effectively zero. For this reason, the optimal covering in a local area should be a BCC lattice, since this is the best known covering in $\mathbb{R}^3$. Since a BCC lattice has a higher covering density than the simplex bound, the optimality gaps presented here leave room for improvement.

4.1. Practical Application

The results presented in Figure 6 demonstrate the evolution of the different methods with increasing size, though all at small sizes. For a practical pattern-indexing application, much larger point sets are needed. Furthermore, whilst the covering radius of a set specifies the maximum error, the distribution of errors is also of practical interest. Figure 7 compares the error histograms of a covering radius optimized set and a cubochoric set, which is used for comparison due to its use in the widely

Table 2
Conjectured optimality gaps for covering radius optimized configurations, with $2I_{60}$ symmetry applied. N is the number of points in each set, θ is the covering radius, θ^* is the covering radius of the simplex bound on $\mathbb{S}^3$, conjectured to be a lower bound (Böröczky, 2004). The optimality gap percentage is $100\,(\theta/\theta^* - 1)$. $^\dagger N = 8$ and $N = 120$ are the point sets containing the vertices of the 16-cell and 600-cell respectively, included here to highlight the tightness of the simplex bound for point sets consisting of regular tetrahedral cells.

N	θ	θ^*	Opt. Gap
$8^\dagger$	60.00°	60.00°	0.00%
$120^\dagger$	22.24°	22.24°	0.00%
1920	9.05°	8.73°	3.68%
3960	7.20°	6.85°	5.05%
6000	6.27°	5.96°	5.07%
7920	5.71°	5.44°	4.95%
9960	5.27°	5.04°	4.67%
12000	4.96°	4.73°	4.71%
13920	4.72°	4.50°	4.76%
15960	4.50°	4.30°	4.54%
18000	4.33°	4.13°	4.74%
19920	4.18°	4.00°	4.61%
24000	3.93°	3.76°	4.72%
27960	3.72°	3.57°	4.31%
31920	3.56°	3.41°	4.38%
36000	3.43°	3.28°	4.47%
39960	3.31°	3.17°	4.62%
43920	3.21°	3.07°	4.67%
48000	3.11°	2.98°	4.49%
60000	2.89°	2.77°	4.35%
79920	2.63°	2.51°	4.48%
99960	2.44°	2.33°	4.64%
139920	2.19°	2.09°	4.83%
180000	2.01°	1.92°	4.84%

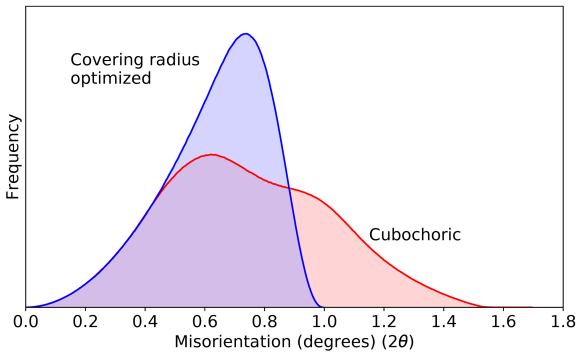


Figure 7
Error histogram for a covering-radius optimized point set with 6×10^6 orientations and a cubochoric set with 6.3×10^6 orientations. Here, both point sets cover the full space of $SO(3)$, which corresponds to indexing a material with a triclinic crystal lattice. The covering-radius optimized point set has a lower maximum error (1.00° vs. 1.72°) and a better overall distribution of errors.

used EMsoft microscopy software (De Graef, 2017). In order to generate the error histogram 10^8 random orientations were sampled; for each sampled orientation, the misorientation is calculated to the nearest orientation in the dictionary set. A KD-tree (Bentley, 1975) is used to quickly find the closest dictionary orientation. In addition to a smaller maximum error, the covering-radius optimized set has a better overall error distribution. This is achieved despite the use of a smaller number of orientations.

The maximum error of the covering radius optimized set is 72% smaller than that of the cubochoric set. In the Euclidean limit $\theta \propto n^{-1/3}$, which suggests that a cubochoric set would require approximately 5 times as many points to achieve the same maximum error.

Using the symmetry relationships described in Section 2.4, we have created orientation sets for every Laue group with maximum misorientations (2θ) of $< 1°, 2°, 3°, 4°$ and $5°$, with optimality gaps less than 6% for every set. The orientation sets available online (Larsen & Schmidt, 2017).

5. Summary

In this work we have shown how the spherical covering problem can be adapted to create nearly optimal orientation sets. As opposed to existing methods which have prioritized fast generation, which have chosen to optimize the maximum error (the covering radius) above all else. The resulting point sets exhibit a smooth decrease in covering radius with an increasing number of points, and have a lower covering radius than any existing method.

In addition to an exact calculation of the covering radius, we have derived an expression for the simplex bound covering radius on $\mathbb{S}^3$, which allows the calculation of conjectured optimality gaps and provides a benchmark for future methods. We have also shown how different symmetry groups can be imposed on the point sets without loss of optimality, which permits integration with the crystallographic fundamental zones of all Laue groups.

The orientation sets we have generated lack the refinable grid-like structure and isolatitudinal properties of other methods, though these properties are fundamentally incompatible with the objective of minimizing the covering radius. Our method also has a high computational requirements; the largest point set requires approximately 4 days of computation time. Nonetheless, we claim that this is a good trade-off, since a point set must only be generated once for each desired error level, and affords a significant performance improvement every time a pattern is subsequently indexed.

Appendix A
Simplex Bound Derivation

A.1. Volume of a Hyperspherical Cap

The volume of a hyperspherical cap in $\mathbb{S}^d$ can be calculated by projection into RF space. Since RF space is radially symmetric about the origin, the projection of a hyperspherical cap with radius θ and centre coordinates $\{1, 0, 0, \ldots, 0\} \in \mathbb{S}^d$ is a sphere with radius $r = \tan(\theta)$ centred at the origin. Thus, the volume of the cap is the radial integral of the product of the surface area of a $(d-1)$-sphere with the RF space density:

$$C_d(\theta) = \int_0^{\tan(\theta)} \frac{S_{d-1}(r)}{(1+r^2)^2} dr \quad \text{where} \quad S_{d-1}(\theta) = \frac{d\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)} \theta^{d-1}$$

For a hyperspherical cap in $\mathbb{S}^3$, this gives:

$$C_3(\theta) = \int_0^{\tan(\theta)} \frac{4\pi r^2}{(1+r^2)^2} dr = \pi(2\theta - \sin(2\theta))$$

This is the same result derived by Moriawiec (Morawiec, 2003; Morawiec, 2010), but without normalization.

A.2. Edge Length of a Regular Spherical Tetrahedron in $\mathbb{S}^3$

Due to the radial symmetry of RF space, the RF projection of a regular spherical tetrahedron with centre coordinates $q_c = \{1,0,0,0\} \in \mathbb{S}^3$ is a tetrahedron with centre coordinates $v_c = \{0,0,0\}$ and vertex coordinates:

$$\begin{array}{ll} v_1 = \{k,k,k\} & v_2 = \{k,-k,-k\} \\ v_3 = \{-k,k,-k\} & v_4 = \{-k,-k,k\} \end{array}$$

From this, we obtain the vertex coordinates in $\mathbb{S}^3$:

$$\begin{array}{ll} q_1 = \frac{1}{\sqrt{1+3k^2}}\{1,k,k,k\} & q_2 = \frac{1}{\sqrt{1+3k^2}}\{1,k,-k,-k\} \\ q_3 = \frac{1}{\sqrt{1+3k^2}}\{1,-k,k,-k\} & q_4 = \frac{1}{\sqrt{1+3k^2}}\{1,-k,-k,k\} \end{array}$$

The circumradius of the tetrahedron is given by the arc length from the centre to any of the vertices:

$$\theta = \arccos\langle q_c, q_i\rangle = \arccos\left(\frac{1}{\sqrt{1+3k^2}}\right) \quad \forall i \qquad (28)$$

The edge length of the tetrahedron is the arc length between any two vertices:

$$l = \arccos\langle q_i, q_j\rangle = \arccos\left(\frac{1-k^2}{1+3k^2}\right) \quad \forall i \neq j \qquad (29)$$

Using Equations (28) and (29) we can express the edge length in terms of the radius:

$$l = \arccos\left(\frac{4\cos^2(\theta)-1}{3}\right) \qquad (30)$$

A.3. Dihedral Angle and Solid Angle of Intersection

Let $\{q_1, q_2, q_3, q_4\}$ be the vertices of a regular hyperspherical simplex in $\mathbb{S}^3$ with the following coordinates:

$$\begin{array}{ll} q_1 = \{1,0,0,0\} & q_2 = \left\{\cos l, -a, \frac{-a}{\sqrt{3}}, z\right\} \\ q_3 = \left\{\cos l, a, \frac{-a}{\sqrt{3}}, z\right\} & q_4 = \left\{\cos l, 0, \frac{-2a}{\sqrt{3}}, z\right\} \end{array}$$

where:

$$a = \sqrt{\frac{1-\cos l}{2}} \qquad z = \sqrt{\sin^2 l - \frac{2}{3}(1-\cos l)}$$

When projected into RF space the tetrahedron has vertices:

$$\begin{array}{ll} v_1 = \{0,0,0\} & v_2 = \frac{1}{\cos l}\left\{-a, \frac{-a}{\sqrt{3}}, z\right\} \\ v_3 = \frac{1}{\cos l}\left\{a, \frac{-a}{\sqrt{3}}, z\right\} & v_4 = \frac{1}{\cos l}\left\{0, \frac{-2a}{\sqrt{3}}, z\right\} \end{array}$$

The dihedral angle of the tetrahedron is then given by:

$$\psi(l) = \arccos\left\langle \frac{v_2 \times v_3}{|v_2 \times v_3|}, \frac{v_2 \times v_4}{|v_2 \times v_4|} \right\rangle = \arccos\left(\frac{\cos l}{2\cos l + 1}\right)$$

Using Equation (30) we can express the dihedral angle in terms of θ:

$$\psi(\theta) = \arccos\left(\frac{4\cos^2(\theta)-1}{8\cos^2(\theta)+1}\right)$$

The solid angle is then given by:

$$\Omega(\theta) = 3\psi(\theta) - \pi = 3\arccos\left(\frac{4\cos^2(\theta)-1}{8\cos^2(\theta)+1}\right) - \pi$$

Since v_1 lies at the origin, this is also the solid angle of intersection of a regular hyperspherical simplex and a hyperspherical cap placed at one of its vertices. We can verify that in the Euclidean limit (where the curvature is zero), $\lim_{\theta\to 0}\Omega(\theta) = 3\arccos\left(\frac{1}{3}\right) - \pi = \arccos\left(\frac{23}{27}\right)$, which is the solid angle for a regular tetrahedron in $\mathbb{R}^3$, and that $\Omega\left(\frac{\pi}{3}\right) = \frac{\pi}{2}$ which is the solid angle of a tetrahedral cell in the 16-cell.

Appendix B
Laue Group Subset Relationships

Table 3
Generators for the seven Laue groups which are subsets of O.

	O	T	D_4	D_2	C_4	C_2	C_1
$\{1,0,0,0\}$	✓	✓	✓	✓	✓	✓	✓
$\{0,0,0,1\}$	✓	✓	✓	✓	✓	✓	
$\{0,1,0,0\}$	✓	✓	✓	✓			
$\{0,0,1,0\}$	✓	✓	✓	✓			
$\left\{\frac{\sqrt{2}}{2},0,0,\frac{\sqrt{2}}{2}\right\}$	✓		✓		✓		
$\left\{\frac{\sqrt{2}}{2},0,0,-\frac{\sqrt{2}}{2}\right\}$	✓		✓		✓		
$\left\{0,\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2},0\right\}$	✓		✓				
$\left\{0,-\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2},0\right\}$	✓		✓				
$\left\{\frac{1}{2},\frac{1}{2},-\frac{1}{2},\frac{1}{2}\right\}$	✓	✓					
$\left\{\frac{1}{2},\frac{1}{2},\frac{1}{2},-\frac{1}{2}\right\}$	✓	✓					
$\left\{\frac{1}{2},\frac{1}{2},-\frac{1}{2},-\frac{1}{2}\right\}$	✓	✓					
$\left\{\frac{1}{2},-\frac{1}{2},-\frac{1}{2},-\frac{1}{2}\right\}$	✓	✓					
$\left\{\frac{1}{2},-\frac{1}{2},\frac{1}{2},\frac{1}{2}\right\}$	✓	✓					
$\left\{\frac{1}{2},-\frac{1}{2},\frac{1}{2},-\frac{1}{2}\right\}$	✓	✓					
$\left\{\frac{1}{2},-\frac{1}{2},-\frac{1}{2},\frac{1}{2}\right\}$	✓	✓					
$\left\{\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2}\right\}$	✓	✓					
$\left\{\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2},0,0\right\}$	✓						
$\left\{\frac{\sqrt{2}}{2},-\frac{\sqrt{2}}{2},0,0\right\}$	✓						
$\left\{\frac{\sqrt{2}}{2},0,\frac{\sqrt{2}}{2},0\right\}$	✓						
$\left\{\frac{\sqrt{2}}{2},0,-\frac{\sqrt{2}}{2},0\right\}$	✓						
$\left\{0,\frac{\sqrt{2}}{2},0,\frac{\sqrt{2}}{2}\right\}$	✓						
$\left\{0,-\frac{\sqrt{2}}{2},0,\frac{\sqrt{2}}{2}\right\}$	✓						
$\left\{0,0,\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}\right\}$	✓						
$\left\{0,0,-\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}\right\}$	✓						

Table 4
Generators for the five Laue groups which are subsets of D_6.

	D_6	D_3	C_6	C_3	C_1
$\{1,0,0,0\}$	✓	✓	✓	✓	✓
$\left\{\frac{1}{2},0,0,\frac{\sqrt{3}}{2}\right\}$	✓	✓	✓	✓	
$\left\{\frac{1}{2},0,0,-\frac{\sqrt{3}}{2}\right\}$	✓	✓	✓	✓	
$\{0,0,0,1\}$	✓		✓		
$\left\{\frac{\sqrt{3}}{2},0,0,\frac{1}{2}\right\}$	✓		✓		
$\left\{\frac{\sqrt{3}}{2},0,0,-\frac{1}{2}\right\}$	✓		✓		
$\{0,1,0,0\}$	✓	✓			
$\left\{0,-\frac{1}{2},\frac{\sqrt{3}}{2},0\right\}$	✓	✓			
$\left\{0,\frac{1}{2},\frac{\sqrt{3}}{2},0\right\}$	✓	✓			
$\left\{0,\frac{\sqrt{3}}{2},\frac{1}{2},0\right\}$	✓				
$\left\{0,-\frac{\sqrt{3}}{2},\frac{1}{2},0\right\}$	✓				
$\{0,0,1,0\}$	✓				

Acknowledgements P.M.L. thanks T.J. Hardin for advice on spherical and hyperspherical harmonics, F. Ram for discussions on the cubochoric method, and N. Wahlberg and E.B. Knudsen for proofreading and helpful suggestions on the manuscript.

References

Adams, B. L., Wright, S. I. & Kunze, K. (1993). *Metall. Mater. Trans.* A**24**(4), 819–831.

Alliez, P., Cohen-Steiner, D., Yvinec, M. & Desbrun, M. (2005). *ACM Trans. Graph.* **24**(3), 617–625.

Altmann, S. L. (2005). *Rotations, Quaternions, and Double Groups*. Courier Dover Publications.

Altschuler, E. L. & Perez-Garrido, A. (2007). *Phys. Rev. E Stat. Nonlin.* **76**(1, 2).

Altschuler, E. L., Williams, T. J., Ratner, E. R., Dowla, F. & Wooten, F. (1994). *Phys. Rev. Lett.* **72**(17), 2671–2674.

Bentley, J. L. (1975). *Communications of the ACM*, **18**(9), 509–517.

Böröczky, K. (2004). *Finite Packing and Covering*, vol. 154. Cambridge University Press.

Böröczky, K (1978). *Acta Math. Acad. Sci. Hung.* **32**(3-4), 243–261.

Brown, R. & Gray, A. (1967). *Comment. Math. Helv.* **42**(3), 222–236.

Chen, L. (2004). *In Proceedings of 13th International Meshing Roundtable*, pp. 109–120.

Chen, Y. H., Park, S. U., Wei, D., Newstadt, G., Jackson, M. A., Simmons, J. P., De Graef, M. & Hero, A. O. (2015). *Microsc. Microanal.* **21**(3), 739–752.

Cohn, H. & Kumar, A. (2007). *J. Am. Math. Soc.* **20**(1), 99–148.

Conway, J. H. & Sloane, N. J. A. (1998). *Sphere Packings, Lattices and Groups*. Springer.

Conway, J. H. & Smith, D. A. (2003). *On Quaternions and Octonions*. Springer New York.

Coxeter, H., Few, L. & Rogers, C. (1959). *Mathematika*, **6**(02), 147–157.

Dahms, M. & Bunge, H. J. (1989). *J. Appl. Cryst.* **22**(5), 439–447.

De Graef, M., (2017). EMsoft - Electron Microscopy Diffraction and Image Computations. `http://github.com/marcdegraef/EMsoft/`.

van Emde Boas, P. (1981). *Another NP-complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice*. Universiteit van Amsterdam. Mathematisch Instituut.

Erber, T. & Hockney, G. (1991). *J. Phys. A-Math. Gen.* **24**(23), L1369–L1377.

Frank, F. (1988). *Metall. Trans. A*, **19**(3), 403–408.

Hardin, D. & Saff, E. (2004). *Notices of the AMS*, **51**(10), 1186–1194.

Hardin, R. H., Sloane, N. J. A. & Smith, W. D., (2017). Spherical codes, book in preparation. `http://neilsloane.com/coverings/`.

He, Y. & Jonas, J. J. (2007). *J. Appl. Cryst.* **40**(3), 559–569.

Karney, C. F. F. (2007). *J. Mol. Graph. Model.* **25**(5), 595–604.

Kuijlaars, A. B. J., Saff, E. B. & Sun, X. (2007). *J. Comput. Appl. Math.* **199**(1), 172–180.

Larsen, P. M. & Schmidt, S., (2017). Spherical Coverings - Data Repository. `http://github.com/pmla/hyperspherical-coverings`.

Li, S. F. & Suter, R. M. (2013). *J. Appl. Cryst.* **46**(2), 512–524.

Mason, J. K. (2009). *Acta Cryst.* **A65**, 259–266.

Mason, J. K. & Schuh, C. A. (2008). *Acta Mater.* **56**(20), 6141–6155.

Morawiec, A. (2003). *Orientations and rotations*. Springer.

Morawiec, A. (2010). *Acta Cryst.* A**66**(6), 717–719.

Morawiec, A. & Field, D. (1996). *Philos. Mag. A.* **73**(4), 1113–1130.

Murakami, J. (2012). *Proc. Am. Math. Soc.* **140**(9), 3289–3295.

Nelder, J. A. & Mead, R. (1965). *Comput. J.* **7**(4), 308–313.

Poulsen, H. F., Nielsen, S. F., Lauridsen, E. M., Schmidt, S., Suter, R., Lienert, U., Margulies, L., Lorentzen, T. & Juul Jensen, D. (2001). *J. Appl. Cryst.* **34**(6), 751–756.

Rakhmanov, E., Saff, E. & Zhou, Y. (1995). *Computational Methods and Function Theory (CMFT 94)*, **5**, 293–309.

Ram, F., Wright, S., Singh, S. & De Graef, M. (2017). *Ultramicroscopy*, **181**, 17–26.

Roşca, D., Morawiec, A. & De Graef, M. (2014). *Modelling Simul. Mater. Sci. Eng.* **22**(7).

Rogers, C. A. (1958). *Proc. London Math. Soc.* **8**(4), 609–620.

Schmidt, S. (2014). *J. Appl. Cryst.* **47**(1), 276–284.

Shoemake, K. (1992). In *Graphics Gems III*, pp. 124–132. Academic Press.

Singh, S. & De Graef, M. (2017). *Microsc. Microanal.* **23**(1), 1–10.

Tóth, L. F. (1964). *Regular Figures*. New York, Paris.

Wales, D. J. & Ulker, S. (2006). *Phys. Rev. B*, **74**(21).

Wright, S. & Nocedal, J. (1999). *Numerical Optimization*, vol. 2. Springer New York.

Yershova, A., Jain, S., LaValle, S. M. & Mitchell, J. C. (2010). *Int. J. Robot. Res.* **29**(7), 801–812.

Paper 7

FELIX: an algorithm for indexing multiple crystallites in X-ray free-electron laser snapshot diffraction images
Beyerlein K R, White T A, Yefanov O, Gati C, Kazantsev I G, Nielsen N F G, Larsen P M, Chapman H N and Schmidt S
Journal of Applied Crystallography
Volume 50, Number 4, Pages 1075-1083

Since my contribution to this paper (c.f. section 11.3) is far from being the main contribution, only the front matter and abstract are included here.

JOURNAL OF APPLIED CRYSTALLOGRAPHY
ISSN 1600-5767

FELIX: an algorithm for indexing multiple crystallites in X-ray free-electron laser snapshot diffraction images

Kenneth R. Beyerlein,[a] Thomas A. White,[a] Oleksandr Yefanov,[a] Cornelius Gati,[a] Ivan G. Kazantsev,[b] Nicolai Fog-Gade Nielsen,[c] Peter M. Larsen,[c] Henry N. Chapman[a] and Søren Schmidt[c]*

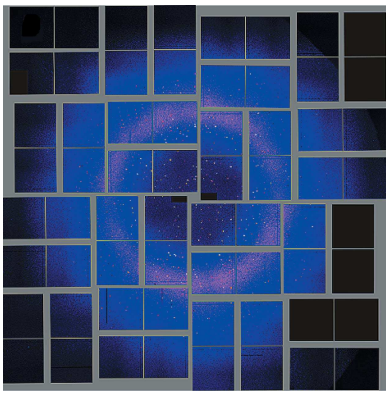
[a]Center for Free-Electron Laser Science, DESY, Notkestrasse 85, 22607 Hamburg, Germany, [b]Institute of Computational Mathematics and Mathematical Geophysics, Lavrentieva 6, 630090 Novosibirsk, Russian Federation, and [c]Department of Physics, Technical University of Denmark, DK-2800, Denmark. *Correspondence e-mail: ssch@fysik.dtu.dk

Received 28 February 2017
Accepted 21 May 2017

Edited by S. Boutet, SLAC National Accelerator Laboratory, Menlo Park, USA

Keywords: serial crystallography; SFX; materials science; structural biology.

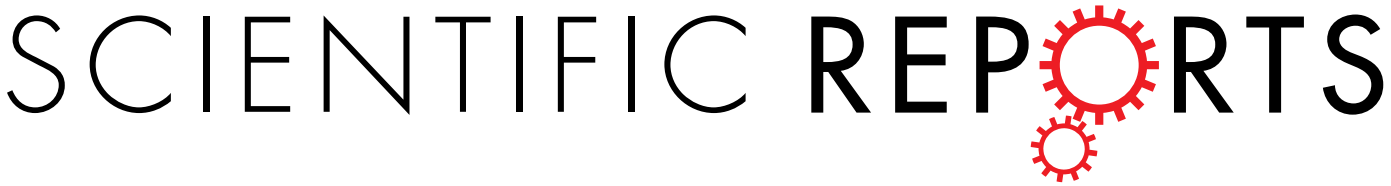
A novel algorithm for indexing multiple crystals in snapshot X-ray diffraction images, especially suited for serial crystallography data, is presented. The algorithm, FELIX, utilizes a generalized parametrization of the Rodrigues–Frank space, in which all crystal systems can be represented without singularities. The new algorithm is shown to be capable of indexing more than ten crystals per image in simulations of cubic, tetragonal and monoclinic crystal diffraction patterns. It is also used to index an experimental serial crystallography dataset from lysozyme microcrystals. The increased number of indexed crystals is shown to result in a better signal-to-noise ratio, and fewer images are needed to achieve the same data quality as when indexing one crystal per image. The relative orientations between the multiple crystals indexed in an image show a slight tendency of the lysozme microcrystals to adhere on $(\bar{1}10)$ facets.



Paper 8

Time-of-Flight Three Dimensional Neutron Diffraction in Transmission Mode for Mapping Crystal Grain Structures
Cereser A, Strobl M, Hall S, Steuwer A, Kiyanagi R, Tremsin A, Bergbäck Knudsen E, Shinohara T, Willendrup P, Bastos da Silva Fanta A, Iyengar S, Larsen P M, Hanashima T, Moyoshi T, Kadletz P M, Krooß P, Niendorf T, Sales M, Schmahl W W, Schmidt S
Scientific Reports
Volume 7, Number 1, Pages 9561

Since my contribution to this paper is, again, far from being the main contribution, and since this paper is tangential to the rest of the thesis, only the front matter and abstract are included here.



OPEN

Time-of-Flight Three Dimensional Neutron Diffraction in Transmission Mode for Mapping Crystal Grain Structures

Received: 25 April 2017
Accepted: 28 July 2017
Published online: 25 August 2017

Alberto Cereser[1,2], Markus Strobl[2,3], Stephen A. Hall[2,4], Axel Steuwer[5,6], Ryoji Kiyanagi[7], Anton S. Tremsin[8], Erik B. Knudsen[1], Takenao Shinohara[7], Peter K. Willendrup[1], Alice Bastos da Silva Fanta[9], Srinivasan Iyengar[2,10], Peter M. Larsen[1], Takayasu Hanashima[11], Taketo Moyoshi[11], Peter M. Kadletz[12], Philipp Krooß[13], Thomas Niendorf[13], Morten Sales[1], Wolfgang W. Schmahl[12] & Søren Schmidt[1]

The physical properties of polycrystalline materials depend on their microstructure, which is the nano- to centimeter scale arrangement of phases and defects in their interior. Such microstructure depends on the shape, crystallographic phase and orientation, and interfacing of the grains constituting the material. This article presents a new non-destructive 3D technique to study centimeter-sized bulk samples with a spatial resolution of hundred micrometers: time-of-flight three-dimensional neutron diffraction (ToF 3DND). Compared to existing analogous X-ray diffraction techniques, ToF 3DND enables studies of samples that can be both larger in size and made of heavier elements. Moreover, ToF 3DND facilitates the use of complicated sample environments. The basic ToF 3DND setup, utilizing an imaging detector with high spatial and temporal resolution, can easily be implemented at a time-of-flight neutron beamline. The technique was developed and tested with data collected at the Materials and Life Science Experimental Facility of the Japan Proton Accelerator Complex (J-PARC) for an iron sample. We successfully reconstructed the shape of 108 grains and developed an indexing procedure. The reconstruction algorithms have been validated by reconstructing two stacked Co-Ni-Ga single crystals, and by comparison with a grain map obtained by post-mortem electron backscatter diffraction (EBSD).

[1]NEXMAP, Department of Physics, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark. [2]European Spallation Source ESS AB, Lund, 22592, Sweden. [3]Niels Bohr Institute, University of Copenhagen, Copenhagen, 2100, Denmark. [4]Division of Solid Mechanics, Lund University, Lund, 22362, Sweden. [5]Nelson Mandela Metropolitan University, Port Elizabeth, 6031, South Africa. [6]University of Malta, Msida, MSD, 2080, Malta. [7]J-PARC center, Japan Atomic Energy Agency, Tokai-mura, 319-1195, Japan. [8]Space Sciences Laboratory, University of California at Berkeley, Berkeley, California, 94720, USA. [9]Center for Electron Nanoscopy, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark. [10]Division of Materials Engineering, Lund University, Lund, 22362, Sweden. [11]Research Center for Neutron Science and Technology, CROSS, Tokai, Naka-gun, 319-1106, Japan. [12]Applied Crystallography and Materials Science, Department of Earth and Environmental Sciences, Ludwig-Maximilians-Universität, München, 80333, Germany. [13]Institut für Werkstofftechnik (Materials Engineering), Universität Kassel, Kassel, 34125, Germany. Correspondence and requests for materials should be addressed to S.S. (email: ssch@fysik.dtu.dk)